



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH

# Smart Wallet,

Sistema de gestió de targetes de contacte.

**Albert Franzi Cros**

Enginyeria en Informàtica

Facultat d'Informàtica de Barcelona

17 Desembre, 2014

Javier Béjar Alonso

Dept. Llenguatges i Sistemes Informàtics

Universitat Politècnica de Catalunya

Agrair als amics i als meus pares que  
han compartit tot el camí d'educació  
al meu costat.

**Títol:** Smart Wallet, Sistema de gestió de targetes de contacte.

**Autor:** Albert Franzí Cros

**Data:** 17 de Desembre de 2014

**Director:** Javier Béjar Alonso

**Departament del director:** Llenguatges i Sistemes Informàtics

**Titulació:** Enginyeria Informàtica

**Centre:** Facultat d'Informàtica de Barcelona (FIB)

**Universitat:** Universitat Politècnica de Catalunya(UPC) - BarcelonaTech

# Índex

1 Introducció al projecte.....	7
1.1 Context.....	7
1.2 Motivació.....	7
1.3 Objectius .....	8
1.4 Organització del document .....	8
2 Estat de l'art .....	10
2.1 Què és una cartera de targetes de contactes?.....	10
2.2 Què és el Reconeixement òptic de caràcters - OCR?.....	11
2.2.1 Esquema bàsic d'un algorisme d'OCR: .....	11
2.2.2 Aplicacions actuals .....	12
2.3 Estudi de sistemes de gestió de contactes.....	12
2.3.1 LinkedIn CardMunch .....	12
2.3.2 ABBYY .....	13
2.3.3 CamCard .....	14
2.3.4 FullContact.....	15
2.3.5 ScanBizCards .....	16
2.3.6 Business Card Reader - Shape.....	17
2.3.7 CardFlick .....	18
2.4 Comparativa - Sistemes de gestió de contactes .....	19
3 Anàlisi de requisits.....	20
3.1 Requisits funcionals .....	20
3.2 Requisits no funcionals .....	21
4 Especificació .....	22
4.1 Situació actual .....	22
4.2 La nova situació .....	22
4.3 Parts interessades.....	23
4.3.1 Empresari.....	23
4.3.2 Professional .....	23
4.3.3 Consumidor .....	23
4.4 Entorn de treball .....	24
4.5 Esdeveniments.....	25
4.6 Especificació casos d'ús de negoci (BUC) .....	26
4.7 Restriccions .....	32
4.8 Casos d'ús de producte .....	33
4.8.1 Diagrama de context .....	33

4.8.2 Limitacions de producte .....	34
4.8.3 Llista de casos d'ús del producte .....	37
4.8.4 Casos d'ús individuals.....	38
5 Disseny .....	44
5.1 Tecnologia .....	44
5.1.1 HTML5 .....	45
5.1.2 CSS.....	45
5.1.3 JavaScript.....	45
5.1.4 AngularJs.....	46
5.1.5 Java.....	48
5.1.6 NodeJs .....	48
5.2 Arquitectura .....	49
5.3 Capa de presentació .....	49
5.4 Capa de domini.....	51
5.5 Capa de dades .....	52
6 Implementació .....	53
6.1 Entorn de desenvolupament.....	53
6.1.1 Git - Gestió de versions .....	53
6.1.2 Sublime Text .....	53
6.1.3 NodeJs + NPM.....	54
6.1.4 MongoDB.....	55
6.1.5 Eclipse ADT .....	56
6.1.6 Nginx .....	56
6.2 Detalls de la implementació.....	58
6.2.1 NodeJs .....	58
6.2.2 Forever .....	61
6.2.3 Passport + Facebook .....	62
6.2.4 Mongoose.....	63
6.2.5 Mocha.....	65
6.2.6 AngularJs.....	66
6.2.7 restApi .....	68
6.2.8 Manifest.xml.....	69
6.2.9 HttpClient .....	70
6.2.10 Realm.....	71
6.2.11 Motor d'OCR .....	73
7 Proves de l'aplicació .....	76
7.1 Servidor .....	76
7.2 Joc de proves .....	76

8 Planificació .....	78
8.1 Fases.....	78
8.1.1 Fase inicial .....	78
8.1.2 Planificació.....	78
8.1.3 Execució.....	78
8.1.4 Tancament.....	78
8.2 Anàlisi de riscos .....	79
8.3 Estimació d'hores.....	80
8.4 Cost econòmic .....	81
9 Conclusions .....	82
10 Treball futur .....	83
10.1 Monetització.....	83
10.2 Comunicació de camp proper.....	83
10.2.1 Què és la tecnologia de Comunicació de camp proper - NFC? .....	83
10.2.2 Estudi de sistemes NFC.....	84
10.2.3 Comparativa - Tecnologia NFC.....	86
10.3 Recomanacions.....	86
11 Bibliografia .....	87
12 Manual d'usuari.....	89
13 Glossari .....	93

# 1 Introducció al projecte

## 1.1 Context

Actualment, segons un **estudi** realitzat per Telefònica, el **54,6%** de la població d'Europa posseeix **SmartPhone**, sent Espanya el país capdavanter amb un **63,2%**. Això implica que més de la meitat de la població té accés a aplicacions mòbil que poden facilitar i/o complementar les tasques de la vida quotidiana.

A més, un **estudi** elaborat al 2011 pel sociòleg i professor de la Universitat de Cornell de Nova York, Trevor Pinch, comenta que un 47% dels usuaris espanyols considera que les aplicacions mòbils milloren les seves vides en l'ús quotidià(44%), viatges(19%) i en el camp del **treball**(11%).

L'estudi esmenta els diversos perfils que podem trobar en els usuaris d'aplicacions i com la tendència està evolucionant de **consum massiu** d'aplicacions cap a un consum més **focalitzat** en les necessitats de l'usuari, on només les aplicacions que poden satisfer una necessitat són les que acaben essent utilitzades. Un altre punt a remarcar de l'estudi és el fet de la **poca fidelitat** que mostren els usuaris en una aplicació. Si existeix una altra aplicació que ofereix millor servei que l'anterior no dubten en canviar.

En la **temàtica** de les targetes de visita o contacte, han evolucionat de significat i contingut. En un principi era un tret característic de la classe alta utilitzades com a element imprescindible d'etiqueta definides per **sofisticades** normes: la convenció social establia que una persona no es presentava a casa d'una altre (a menys que no hagués estat convidada) sense deixar primer la seva targeta de visita. A més, quan s'anava de visita i els propietaris no hi eren es deixava la targeta al servei com a mostra de que s'havia passat pel domicili.

Però el temps i la societat han anat canviant i actualment l'ús de la targeta no és un tret exclusiu de la classe alta o mitjana, sinó que és utilitzada en totes les capes socials en les quals són essencials els detalls del contacte incloent: nom, càrrec, nom de l'empresa, direcció de correu electrònic, mòbil; i recentment, amb l'aparició de les xarxes socials, l'usuari de Twitter o LinkedIn.

Així doncs, donat el context de l'ús de les aplicacions mòbils i de les targetes de visita, a continuació es presenta una plataforma de gestió de targetes mitjançant l'ús de la tecnologia mòbil, i com s'han de complir els diversos objectius previstos i les conseqüències que aportarà la implementació d'aquest.

## 1.2 Motivació

A la vida quotidiana és freqüent anar a restaurants, botigues, dentistes i perruqueries, entre altres, i totes elles solen oferir una targeta de contacte per facilitar la transmissió de dades i poder trucar o consultar informació en qualsevol altre moment. També entre visites, entrevistes i reunions en el món professional s'acostuma a intercanviar targetes corporatives.

Així doncs, al llarg de l'any anem acumulant diverses targetes de diferents àmbits i les anem guardant ja sigui al calaix de l'escriptori o en un directori de contactes.

Els problemes sorgeixen quan volem consultar les targetes i no tenim el directori a mà, o volem compartir la informació amb terceres persones i només disposem d'una targeta. També sol costar organitzar-les correctament o saber com buscar en elles.

### 1.3 Objectius

L'objectiu principal del projecte és la creació d'un sistema de gestió de targetes amb plataforma web i mòbil orientat a facilitar la gestió, consulta i compartició de targetes de contactes en temps real.

Amb la realització del projecte es vol assolir la disponibilitat d'un servei de consulta d'informació dels contactes emmagatzemats a la xarxa mitjançant la connexió online i assegurar l'accès a aquesta informació en offline en cas de no disposar de connexió. A més, el projecte ha de constar de digitalització de targetes físiques via un motor de Reconeixement Òptic de Caràcters (OCR).

S'haurà d'estudiar les funcionalitats de sistemes existents que cobreixen unes necessitats semblants i quines tecnologies serien les adients per a la correcta execució del projecte.

Així doncs, els objectius principals del projecte serien els següents:

- Garantir la **disponibilitat** del servei de consulta d'informació dels contactes emmagatzemats a la xarxa mitjançant la connexió online.
- Assegurar la consulta d'informació dels contactes propis en mode **offline** en cas de caiguda temporal del servidor o mancança d'itinerància de dades. Així doncs, l'usuari haurà de disposar de la funcionalitat d'accedir a tota la informació relativa al seu compte que hagi estat sincronitzada en la última connexió realitzada amb satisfacció amb el servidor.
- Gestió de les targetes adquirides en **directoris** i etiquetes per facilitar la cerca i comunicació en posteriors usos.
- **Digitalització** del contingut de targetes físiques i posterior emmagatzematge de la informació en el dispositiu mòbil i en xarxa.
- Sincronització correcta entre les dades del servidor i les disponibles en local al mòbil mitjançant la connexió a Internet via wifi o itinerància de dades, realitzada durant la interacció de l'usuari amb l'aplicació o quan aquesta connexió és satisfactoria i la informació està desactualitzada.
- **Compartició** de les targetes de contacte entre els diversos usuaris del sistema.

### 1.4 Organització del document

Aquest document descriu el sistema desenvolupat per a la correcta satisfacció de les necessitats detectades en el marc de la gestió de targetes de contacte.

El projecte és exposat de la següent manera:

**Estat de l'art** Conté un estudi exhaustiu dels sistemes existents i de les tecnologies utilitzades per avaluar el camí a seguir en el desenvolupament del projecte.



**Planificació** Informació de l'ordre de les tasques a seguir i l'anàlisi dels riscos. Conté l'avaluació de l'execució.

**Anàlisi de requisits** Comprensió del propòsit i funcions del sistema. Conté les característiques del sistema i la descripció de les funcionalitats que satisfan els objectius del sistema.

**Especificació** dels diversos casos d'ús i de negoci que conformen el sistema.

**Disseny** Descriu el disseny del sistema a nivell d'arquitectura i de tecnologia.

**Implementació** Conté informació tècnica del sistema, la configuració de l'entorn de desenvolupament i els detalls de l'implementació.

**Testeig** Jocs de prova i metodologia utilitzada per comprovar el correcte funcionament del sistema.

**Conclusió** Anàlisi final del transcurs del projecte així com dels resultats obtinguts.

**Treball futur** Propostes a implementar un cop finalitzat el projecte.

**Bibliografia** Conté material addicional que s'ha utilitzat en l'execució del projecte.

## 2 Estat de l'art

A continuació es presenten els diversos **punts clau** que conformen el projecte i dels quals es procedeix a estudiar en detall les tecnologies i sistemes existents en el mercat que **satisfan** les necessitats esmentades.

### 2.1 Què és una cartera de targetes de contactes?

En el món empresarial és costum presentar-se als nous contactes mitjançant l'entrega d'una targeta de contacte on es disposa tota la **informació** referent a aquest com a finalitat d'estar disponible en qualsevol moment pel correcte oferiment del servei representat.

Aquestes targetes consten, entre d'altres, del nom del contacte, telèfon fix i/o mòbil, correu corporatiu, càrrec, logotip de l'empresa representada i l'adreça de l'oficina. No obstant, les empreses solen oferir, a més, targetes d'informació referent a l'empresa sense delegar cap persona.

Així doncs, al llarg del temps, una persona va guardant targetes de contacte fins a sorgir la necessitat de gestionar-les degut al **volum que ocupen**.

Inicialment, la gestió es duu a terme mitjançant un directori de targetes físic, el qual sol disposar de ranures per anar introduint-les de la manera més adient. Més tard, paral·lelament s'inicia la **virtualització** de les **dades** de les targetes en diversos sistemes d'informació per gestionar-les.

## 2.2 Què és el Reconeixement òptic de caràcters - OCR?

El *Reconeixement Òptic de Caràcters*, així com el reconeixement de text, en general són aplicacions dirigides a la **digitalització de textos**. Identifiquen automàticament **símbols** o **caràcters**, que pertanyen a un determinat alfabet, a partir d'una imatge per emmagatzemar-ho en forma de dades amb les qual podem interactuar amb un programa d'edició de text o similars.

El procés bàsic que es du a terme en el OCR és convertir el text que apareix en una imatge en dades que podran ser editades i utilitzades com a tal per qualsevol programa o aplicació que les necessiti.

Partint d'una imatge perfecta, és a dir, una imatge de dos nivells de gris, el reconeixement d'aquests caràcters es realitzarà bàsicament comparant-los amb uns **patrons** o plantilles que contenen tots els possibles caràcters. Ara bé, les imatges reals no són perfectes, per tant el OCR es troba amb diversos problemes:

- El dispositiu que obté la imatge pot introduir nivells de grisos al fons que no pertanyen a la imatge original.
- La resolució d'aquests dispositius pot introduir soroll a la imatge, afectant als píxels que han de ser processats.
- La separació dels caràcters, al no existir un espai fix entre ells, pot produir errors a l'hora del reconeixement.
- La connexió de dos o més caràcters per píxels comuns també pot produir errors.

### 2.2.1 Esquema bàsic d'un algorisme d'OCR:

Tots els algorismes d'OCR es basen en els següents quatre passos:

- **Binarització:** Conversió de la imatge a blanc i negre obtenint com a resultat una imatge on queden clarament marcats els contorns dels caràcters i símbols que conté la imatge.
- **Fragmentació o segmentació de la imatge:** Implica la detecció, mitjançant procediments d'etiquetat determinista o estocàstic, dels contorns o regions de la imatge, basant-se en la informació d'intensitat o informació espacial. Així doncs, es permet la descomposició d'un text en diferents entitats lògiques.
- **Simplificació dels components:** Consisteix en anar esborrant successivament els punts dels contorns de cada component, de forma que es preservi la seva tipologia. S'ha de fer un escombrat en paral·lel, és a dir, senyalar els píxels esborrables per eliminar-los tots a la vegada.
- **Comparació amb patrons:** Es comparen els caràcters obtinguts anteriorment amb uns caràcters teòrics (patrons) emmagatzemats en una base de dades.

## 2.2.2 Aplicacions actuals

Des de l'aparició dels algorismes d'OCR han estat molts els àmbits que han introduït aquests processos per augmentar el rendiment i oferir nous serveis. A continuació s'exposen tres aplicacions de les més destacades:

- **Reconeixement de texts manuscrits.**
- **Reconeixement de matrícules:** Aplicacions utilitzades en els radars que han de ser capaços de localitzar la matrícula d'un vehicle en condicions d'il·luminació, perspectiva i entorn variables.
- **Indexació en bases de dades:** Es basa en la utilització de cercadors, **DIRS**, que proporcionen la possibilitat de cercar imatges mitjançant el text contingut en aquestes. Mitjançant un algorisme d'OCR extreu el text que apareix en les imatges i l'utilitza com a metadata que podrà ser utilitzada en les cerques.

## 2.3 Estudi de sistemes de gestió de contactes

A continuació es detalla el conjunt de sistemes informàtics existents que satisfan la necessitat de gestionar les targetes de contacte.

### 2.3.1 LinkedIn CardMunch

<a href="http://www.cardmunch.com">www.cardmunch.com</a>	Adreça web
LinkedIn Corporation	Empresa gestora
Gener 2011 ( <i>Última actualització Dec-2011</i> )	Data de llançament
Internacional	Àmbit polític
Negocis	Sectors
225 milions d'usuaris al món, dels quals 5 milions espanyols.	Quota de mercat

#### Serveis

Ofereix la possibilitat d'emmagatzemar les targetes de contactes en una aplicació mòbil mitjançant OCR. Es disposa d'un llistat de contactes ordenats alfabèticament i dels quals es mostra la informació en camps separats i una imatge de la targeta escanejada. De manera addicional permet contactar via LinkedIn en cas de que el mail del contacte estigui registrat anteriorment.

#### Objectius

Proveir una correcta gestió dels contactes.

#### Forma de negoci

CardMunch ofereix el servei de manera gratuïta al tractar-se d'una funcionalitat afegida al sistema d'informació de LinkedIn.

### 2.3.2 ABBYY

<a href="http://www.abbyy.com">www.abbyy.com</a>	Direcció web
ABBYY	Empresa gestora
Internacional	Àmbit polític
Negocis	Sectors
Android: 10.000 a 50.000 usuaris. IOS: 50.000 a 75.000 usuaris.	Quota de mercat

#### Serveis

L'aplicació ofereix les següents funcionalitats:

- Obtenció de dades automàtica mitjançant la tecnologia OCR. L'aplicació pot obtenir els següents camps: nom complet de la persona, nom de la companyia, càrrec, número de mòbil, correu electrònic i pàgina web. A més, emmagatzema la imatge original.
- 21 llenguatges disponibles per a la captura i postprocés de la informació.
- Compartició de targetes mitjançant WIFI entre diversos usuaris que continguin la mateixa aplicació.
- Nombre il·limitat de targetes a emmagatzemar.

#### Objectius

Proveir una correcta gestió dels contactes alhora que es facilita la compartició d'aquests amb altres contactes.

#### Forma de negoci

ABBYY BCR està disponible en les següents plataformes:

Plataforma	Pro	Lite
<i>Android</i>	€7.99	<i>Free</i>
<i>IOS</i>	\$89.99	\$4.99
<i>Windows</i>	€24.95	-

### 2.3.3 CamCard

<a href="http://www.camcard.com">www.camcard.com</a>	Direcció web
CamCard	Empresa gestora
Internacional	Àmbit polític
Negocis	Sectors
Android: 1.000.000 a 5.000.000 d'usuaris.	Quota de mercat

#### Serveis

L'aplicació ofereix les següents funcionalitats:

- Obtenció de dades automàtica mitjançant la tecnologia OCR. Autodetecció de l'orientació del text i de la targeta.
- 12 llenguatges disponibles per a la captura i postprocés de la informació.
- Addició de comentaris i arxius a les targetes, accés directe a trucades i enviament de correus electrònics mitjançant la propia app.
- Codi de seguretat per a la cartera de targetes.
- Sincronització entre els diversos dispositius del mateix usuari.

#### Objectius

Proveir una correcta gestió dels contactes alhora que es facilita la compartició d'aquests entre els diversos dispositius dels quals es disposa i satisfà la necessitat de prendre notes sobre els determinats contactes.

#### Forma de negoci

CamCard està disponible en totes les plataformes (Android, IOS, BlackBerry, Windows). Permet l'escaneig de 100 targetes en el mode Premium. Després el cost és de \$2.99 en els països occidentals i de \$11.99 a Àsia.

### 2.3.4 FullContact

<a href="http://www.fullcontact.com">www.fullcontact.com</a>	Direcció web
FullContact	Empresa gestora
Internacional	Àmbit polític
Negocis	Sectors

#### Serveis

L'aplicació ofereix les següents funcionalitats:

- Obtenció de dades automàtica mitjançant la tecnologia OCR.
- Creació de llistes i possibilitat de compartir-les amb altres contactes.
- Sincronització de contactes: Importació i exportació d'altres agendes de contactes (Gmail, Facebook, LinkedIn i Twitter).
- Detecció i fusió de contactes duplicats.

#### Objectius

Proveir una correcta gestió dels contactes alhora que es facilita la compartició d'aquests entre els diversos contactes dels quals es disposa. Centralitzar les diverses agendes de contactes en una mateixa.

#### Forma de negoci

FullContact actualment està en versió **Beta**, disposa de la versió escriptori de manera gratuïta, exceptuant l'aplicació d'escaneig de targetes la qual té un cost de \$9.99/mes o de \$99.99/any.

### 2.3.5 ScanBizCards

<a href="http://www.scanbizcards.com">www.scanbizcards.com</a>	Direcció web
ScanBiz Mobile Solutions LLC.	Empresa gestora
Internacional	Àmbit polític
Negocis	Sectors

#### Serveis

L'aplicació ofereix les següents funcionalitats:

- Obtenció de dades automàtica mitjançant la tecnologia OCR amb suport a 23 llenguatges.
- Exportació del conjunt de contactes a un **CRM**.
- Enviament d'invitacions de contacte mitjançant LinkedIn.
- Sincronització dels contactes en un compte personal al cloud.
- Integració d'Evernote.

#### Objectius

Proveir una correcta gestió i sincronització dels contactes en el cloud. A més, satisfà la connexió amb un CRM.

#### Forma de negoci

ScanBizCards disposa d'una l'aplicació Lite per a Android de manera gratuïta i d'aplicacions de pagament per IOS (\$2.99) i Windows8 (\$3.99).



### 2.3.6 Business Card Reader - Shape

<a href="http://www.shape.ag">www.shape.ag</a>	Direcció web
SHAPE	Empresa gestora
Internacional	Àmbit polític
Negocis	Sectors

#### Serveis

L'aplicació s'encarrega d'escanejar les targetes de contactes i introduir el contingut en els corresponents camps de text. A més, un cop introduït el contacte ofereix la possibilitat de cercar el contacte a la xarxa de LinkedIn.

#### Objectius

L'objectiu principal de l'aplicació és digitalitzar el contingut de les targetes de contacte.

#### Forma de negoci

SHAPE ofereix l'aplicació estàndard a \$4.99 i una versió ampliada que dona suport a Xinès, Japonès i Coreà per \$6.99.

### 2.3.7 CardFlick

<a href="http://www.cardflick.co">www.cardflick.co</a>	Direcció web
CardFlick	Empresa gestora
Internacional	Àmbit polític
Negocis, Disseny	Sectors

#### Serveis

La funcionalitat principal de l'aplicació és el disseny i creació d'una targeta de negoci de contacte la qual poder compartir posteriorment. A més també ofereix la possibilitat de tenir una llista de contactes.

Un altre servei que ofereix és tenir les dades sincronitzades al cloud.

Actualment només disposa de l'aplicació per IOS.

#### Objectius

L'objectiu principal de l'aplicació és la creació de targeta personal i la compartició d'aquesta.

#### Forma de negoci

CardFlick ofereix l'aplicació de manera totalment gratuïta.

## 2.4 Comparativa - Sistemes de gestió de contactes

A continuació es prossegueix a presentar la comparativa de les diverses funcionalitats i aspectes d'interès que s'han observat en els anàlisi de les empreses i projectes anteriors referents a la gestió de contactes.

S'ha observat com la funcionalitat principal era garantir la satisfacció de la necessitat de gestionar les targetes de contactes en una agenda de contactes i la correcta **digitalització** d'aquestes provinents de diferents **llenguatges** (de 18 a 22 llengües diferents).

Les diverses aplicacions captaven la informació mitjançant la utilització de la tecnologia **OCR**, agilitzant la introducció de les dades als diversos camps de text d'interès. A més, algunes d'aquestes oferien la possibilitat de confirmar la informació mitjançant la xarxa **LinkedIn** o simplement la funcionalitat extra de poder **connectar** amb la plataforma.

Algunes d'elles permeten la compartició dels contactes amb altres usuaris o dispositius, però sempre dins de la mateixa plataforma i sempre unidireccional amb permís del enviant.

Un altre fet important és la **sincronització** dels contactes a un **cloud** per garantir la permanència i seguretat d'aquests, eliminant així el temor a perdre el conjunt de contactes emmagatzemats de manera local.

Un punt a comentar és que totes les aplicacions estaven orientades al nínxol empresarial, on els comercials són els principals clients potencials.

Finalment, una funcionalitat important que només contempla ScanBizCards és la incorporació i compatibilitat amb un sistema d'informació de gestió de relació amb els clients (**CRM**), fet que a una companyia amb un **Sistema de planificació de recursos empresarials** (ERP) implementat podria interessar de manera satisfactòria.

## 3 Anàlisi de requisits

Els requisits són el conjunt d'accions i/o funcionalitats que els usuaris esperen que faci el sistema. Així doncs, s'ha de detallar el problema de manera simplificada i detallada, definint el **què** del problema i, a continuació, en el disseny (*secció x pàgina y*) definir el **com** de la solució.

### 3.1 Requisits funcionals

- **Registre d'usuari:** El sistema ha d'oferir l'opció de registre de nous usuaris.
- **Identificació d'usuari:** El sistema ha de permetre la identificació correcta de l'usuari i l'accés a la informació pertanyent a aquest.
- **Desconnexió:** L'usuari ha de poder desconnectar el dispositiu del sistema.
- **Inserció de nova targeta:** L'usuari ha de poder registrar al sistema les dades d'una nova targeta de contacte.
- **Consulta de directoris de targetes:** El sistema ha de proveir a l'usuari dels directoris amb les corresponents targetes a les que dit usuari té accés.
- **Afegir nota:** L'usuari podrà registrar notes en les targetes del sistema per a una futura consulta.
- **Consultar notes:** El sistema ha de permetre la consulta de les notes registrades en una targeta.
- **Enllaçar xarxa social:** Autoritzar al sistema a cercar contactes registrats en el sistema que també estan a la xarxa social de l'usuari.
- **Veure llistat d'amistats:** El sistema ha de mostrar el conjunt d'usuaris registrats que estan enllaçats amb l'usuari principal mitjançant alguna de les xarxes socials inserida anteriorment.
- **Dades sincronitzades:** El sistema ha de mantenir les dades actualitzades amb el servidor i permetre als dispositius mòbils actualitzar-se al tenir connexió activa amb aquest.
- **Esborrar directoris, targetes i notes:** El sistema ha de permetre eliminar directoris, targetes i notes registrades per l'usuari.

## 3.2 Requisits no funcionals

A continuació es presenten el conjunt de requisits no funcionals que complementen els requisits funcionals esmentats anteriorment:

- **Interfície amigable:** La interfície del sistema ha de ser fàcil d'utilitzar i les funcionalitats intuïtives de comprendre donat que el sistema serà utilitzat per usuaris no experts i la seva satisfacció dependrà en part de la facilitat de l'ús del sistema.
- **Temps de resposta:** El sistema ha de garantir una resposta àgil. Ha de facilitar la informació o la transició d'aquesta de manera ràpida i fluida.
- **Fiabilitat:** El sistema ha de permetre l'actualització de l'aplicació per corregir els possibles errors que s'hi puguin trobar.
- **Quantitat d'usuaris:** El sistema ha de suportar el creixement del nombre d'usuaris i per tant un augment de la mida total de la informació registrada.
- **Extensibilitat de les funcionalitats:** El sistema ha de permetre la correcta integració de noves funcionalitats. L'increment d'usuaris o l'aparició de nous requisits poden comportar la necessitat d'integració de nous mòduls.

## 4 Especificació

### 4.1 Situació actual

En el context actual, els treballadors que acostumen a tenir contacte amb altres empreses realitzen un intercanvi de targetes de contacte al final de les reunions per tal de conservar la informació dels altres assistents.

Per altra banda, els possibles usuaris disposen de diverses aplicacions en les que emmagatzemen la informació referent a contactes al mòbil. A més, tal com s'ha detallat anteriorment en *la secció 2.Estat de l'art*, només alguns ofereixen la possibilitat de compartir en xarxa local o sincronitzar-ho amb una base de dades remota.

Per a la digitalització de dades de les targetes de contactes s'utilitza la tecnologia OCR implementada en algunes de les aplicacions disponibles en el mercat la qual sovint requereix de la confirmació camp a camp per part de l'usuari.

També existeixen algunes aplicacions amb la possibilitat de realitzar anotacions als contactes. Tot i això, la vesant de xarxa social no està explotada i la compartició de contactes, anotacions i recomanacions no es troba satisfeta.

### 4.2 La nova situació

Actualment, una gran part de la població disposa de «*SmartPhone*» i realitza l'ús d'aplicacions en la vida quotidiana.

Accions repetitives, quotidianes i manuals són sovint reemplaçades per aplicacions que satisfan correctament dites necessitats a més d'oferir funcionalitats extres.

Així doncs, la proposta de projecte exposada a continuació consistirà en reutilitzar tecnologies existents en els mòbils per tal d'aportar un valor afegit vers les aplicacions actuals:

- Digitalització de targetes mitjançant tecnologia OCR.
- Validació de dades introduïdes mitjançant OCR i/o manualment mitjançant la cerca i comprovació dels contactes ja existents en el sistema.
- Creació i gestió de directoris de targetes.
- Etiquetatge de targetes.
- Cerca de targetes i directoris en el sistema.
- Sincronització amb el sistema.

Compartició de directoris i targetes amb altres usuaris del sistema.

## 4.3 Parts interessades

En el següent apartat es presenten els diversos actors que poden interactuar amb el sistema i quins objectius esperen satisfer.

### 4.3.1 Empresari

Entitat encarregada de supervisar i gestionar les accions executades pel conjunt de treballadors que conformen l'empresa. També s'encarrega de la gestió del personal i del corresponent compliment de les seves funcions. La seva activitat principal és acumular riquesa mitjançant el subministrament de serveis i/o productes a clients.

#### Objectius

- Promocionar el negoci.
- Facilitar dades correctes i actualitzades de l'empresa (adreça, correu electrònic, persones de contacte, etc).

### 4.3.2 Professional

Entitat que executa un conjunt de funcions dins del marc legal d'una empresa.

#### Objectius

- Facilitar i comunicar la informació de contacte a terceres persones.
- Mantenir actualitzada la informació del perfil empresarial (càrrec personal, correu corporatiu, telèfon d'empresa, empresa).
- Obtenir feedback dels clients.

### 4.3.3 Consumidor

Entitat que obté una satisfacció d'una determinada necessitat al consumir i/o fer ús dels serveis d'una empresa. A més, sovint guarda informació de contacte sobre les empreses, que han prestat servei o possibles empreses potencials que poden prestar servei en un futur, per poder consultar i/o compartir aquesta informació per a contactar amb aquestes.

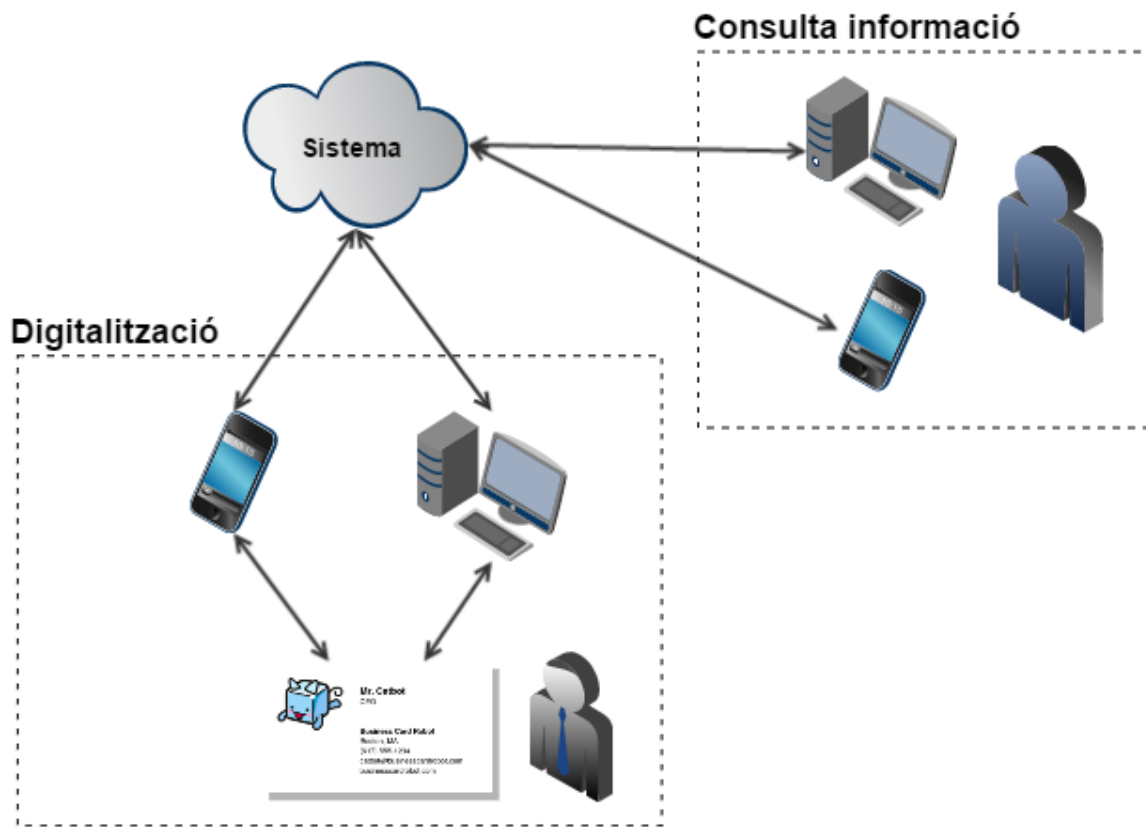
#### Objectius

- Adquirir targetes de contacte de professionals i empreses.
- Gestionar en directoris i etiquetes les diverses targetes per a una correcta administració.
- Consulta en tot moment de la informació de les targetes disponibles.
- Compartir targetes de contacte.

Així doncs, podem distingir tres tipologies de parts interessades que alhora poden trobar-se en una mateixa persona. És a dir, les necessitats a satisfer d'un usuari poden ser qualsevol combinació de les diverses parts presentades anteriorment.

També podríem generalitzar els actors Empresari i Professional com subconjunts de l'actor Consumidor.

## 4.4 Entorn de treball



Imatge 1: Entorn de treball

L'entorn de treball defineix els límits entre el sistema i l'entorn. En el nostre cas tenim que la interacció de l'usuari es realitza mitjançant dispositius mòbils i dispositius amb navegadors web.

Per un costat disposem del bloc de digitalització que consta del traspàs d'informació de targetes al sistema, i del bloc de consulta d'informació on l'usuari gestionarà la informació rebuda pel sistema.



## 4.5 Esdeveniments

A continuació es detalla el conjunt d'esdeveniments com a interacció dels actors amb el sistema:

Nom	Entrada	Sortida	Resum del BUC
1) Introducció d'una targeta	Informació referent a una targeta de contacte	Targeta creada correctament	L'usuari introdueix la informació adient a una targeta de contacte.
2) Sol·licitud de sincronització	Interrupció del rellotge	Resultat de la sincronització	El rellotge del sistema realitza una interrupció en el dispositiu mòbil per a sincronitzar amb el sistema d'informació.
3) Etiquetatge de targetes	Sol·licitud d'etiquetatge	Targeta correctament etiquetada	S'afegeix una etiqueta a la descripció de la targeta.
4) Cerca de targetes	Sol·licitud amb conjunt de paràmetres que acoten la cerca	Conjunt de targetes	El sistema retorna el conjunt de targetes existents amb informació relativa als paràmetres de cerca.
5) Consulta d'una targeta	Sol·licitud d'una targeta	Targeta	El sistema retorna la informació de la targeta seleccionada
6) Consulta contactes	Sol·licitud de contactes	Targes de contactes	L'usuari consulta el conjunt de contactes del sistema
7) Creació de nou directori	Sol·licitud de creació de nou directori	Resultat de la petició i directori	L'usuari realitza la petició de crear un nou directori de targetes en el sistema.
8) Escaneig de targeta	Imatge de la targeta	Informació continguda en la targeta	Es realitza la captació d'informació continguda en la targeta de contacte física.
9) Consulta de directori	Sol·licitud d'un directori	Directorio amb el conjunt de targetes contingudes	El sistema retorna la informació del directori sol·licitat amb el conjunt de targetes que el conformen.
10) Importació d'una targeta via el sistema	Targeta compartida i usuari receptor	Resultat de la importació	L'usuari sol·licita la importació de la targeta d'un altre usuari del sistema.

## 4.6 Especificació casos d'ús de negoci (BUC)

A continuació es presenten els diversos diagrames d'activitat corresponents als esdeveniments detallats en l'apartat anterior:

### Consultar targeta

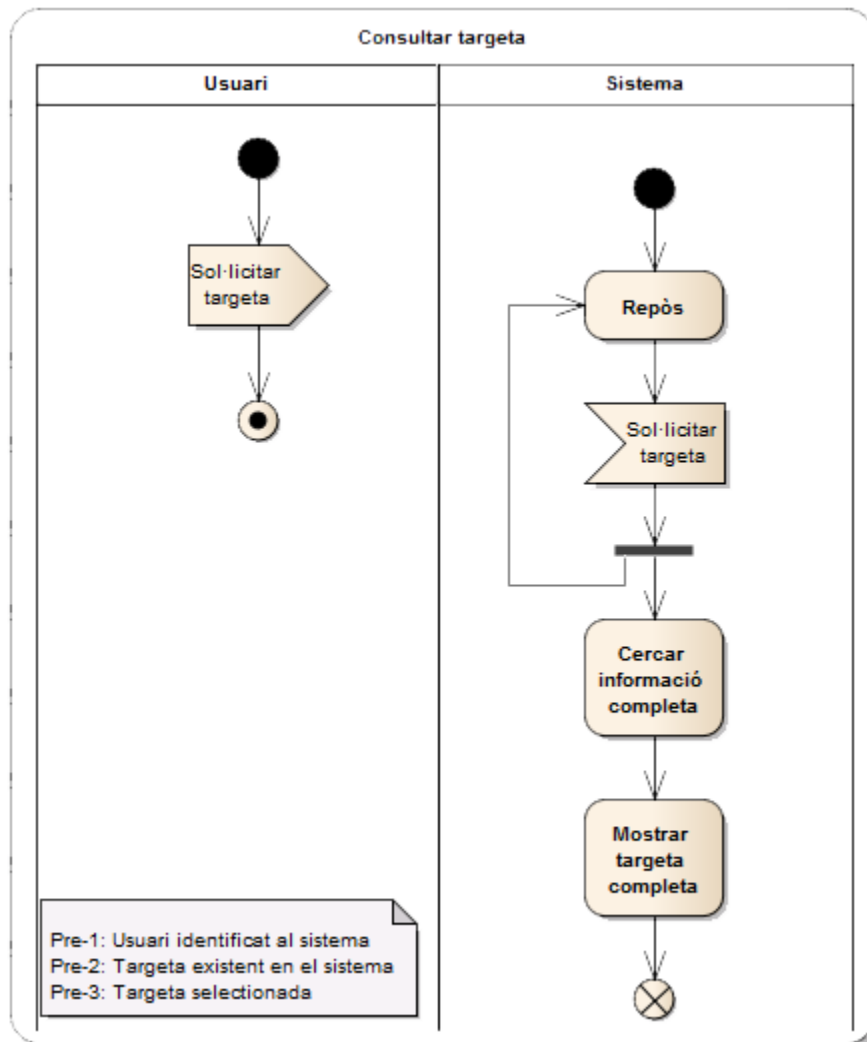


Diagrama 1: Consultar targeta

Quan un usuari sol·liciti la informació d'una targeta específica, el sistema rebrà l'esdeveniment i retornarà la informació respectiva de la targeta sol·licitada.

## Escaneig targeta

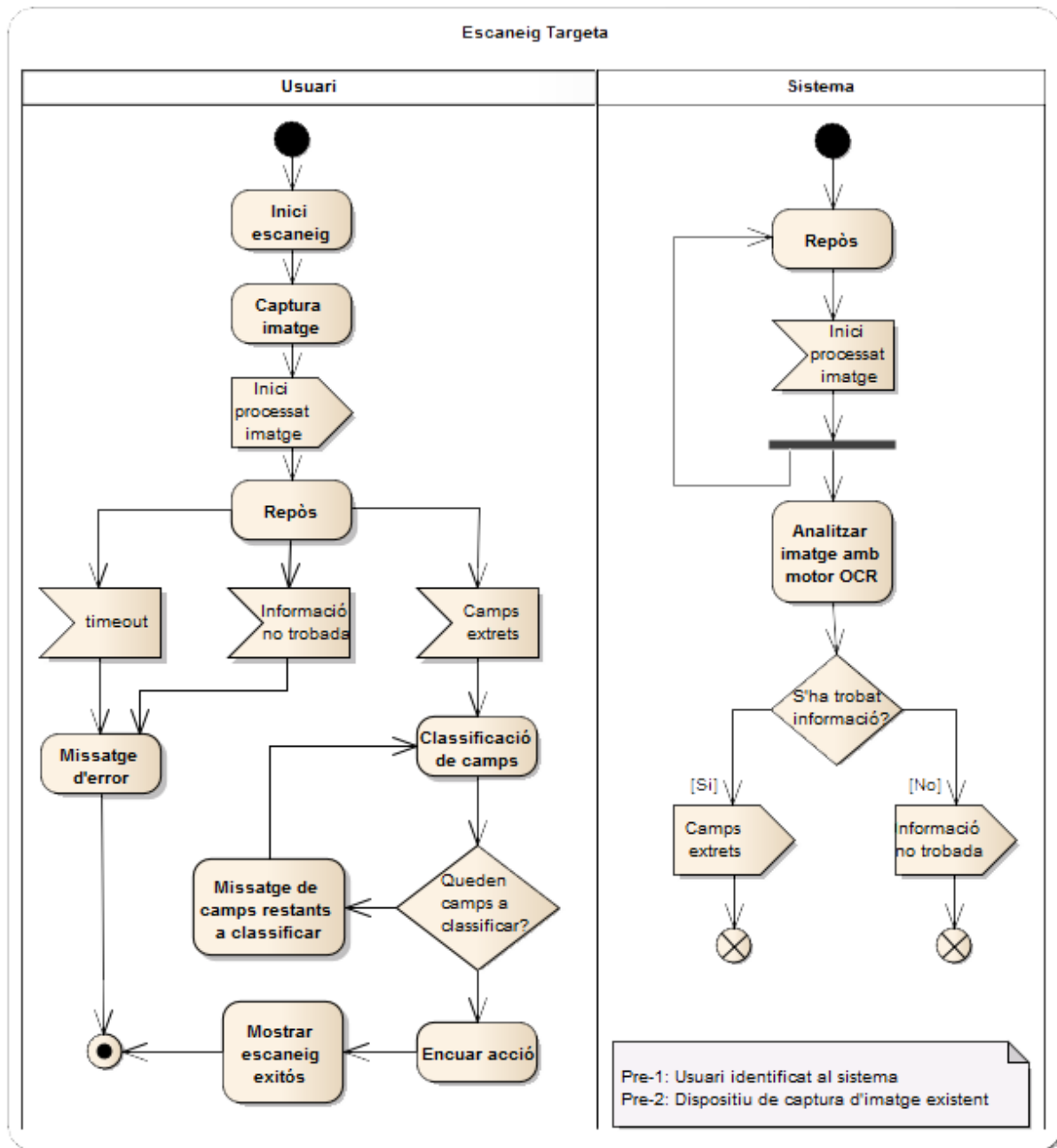


Diagrama 2: Escaneig Targeta

L'usuari inicia l'escaneig de la targeta amb la captura de la imatge. El sistema rebrà la imatge a processar i intentarà analitzar-la per extreure el text. En cas afirmatiu retornarà el text corresponent, sinó, informarà que no hi ha informació.

L'usuari al rebre el text extret el classificarà en el camp corresponent. Finalment es registrarà la nova targeta al sistema.

## Cerca de targetes

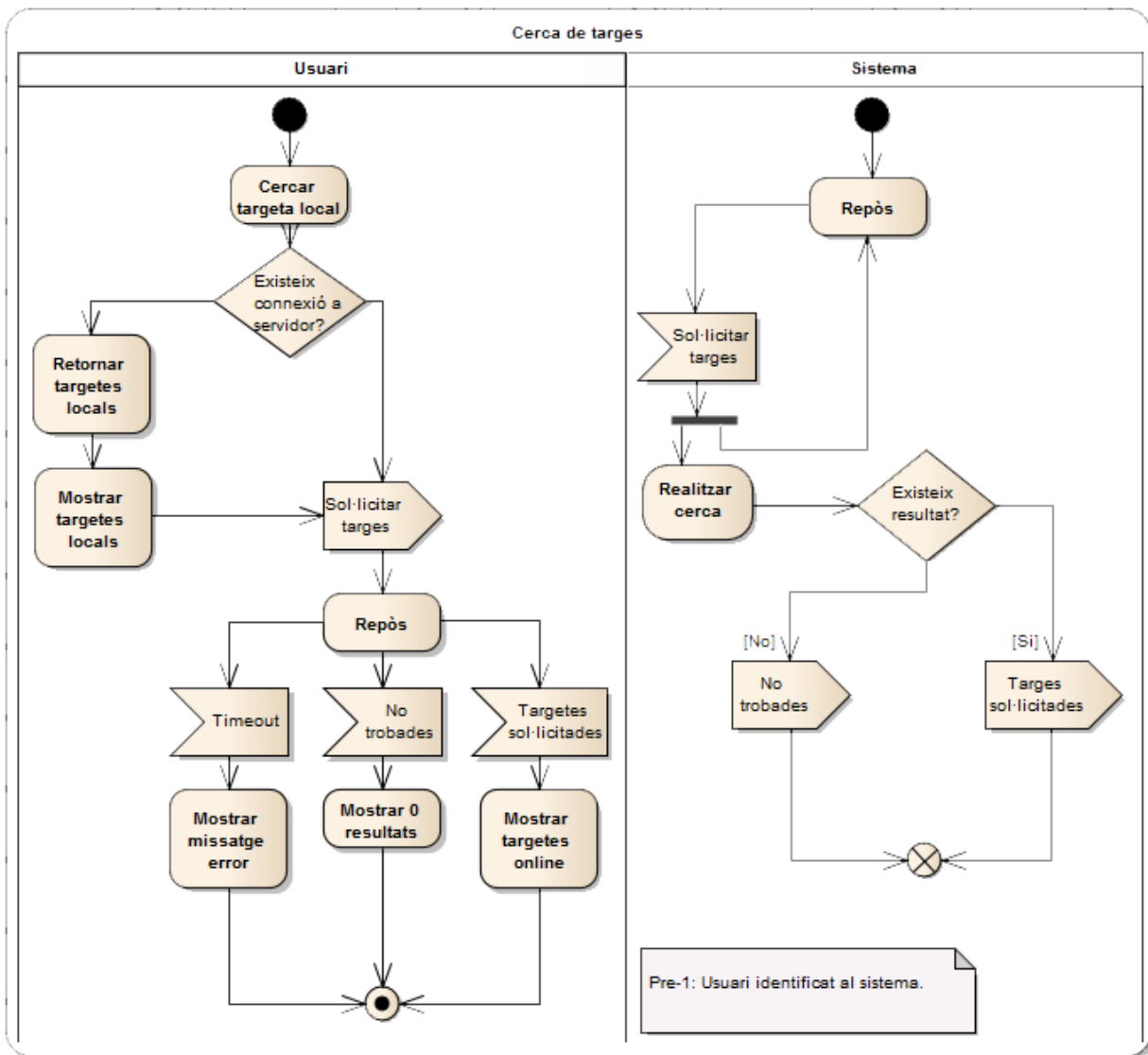


Diagrama 3: Cerca de targetes

L'usuari executarà una cerca al sistema, en cas de no existir connexió amb el servidor el sistema mirarà si disposa de la informació de manera local i retornarà el conjunt de targetes coincidents amb la cerca.

En cas d'existir connexió, el sistema farà la consulta al servidor.

## Creació nou directori

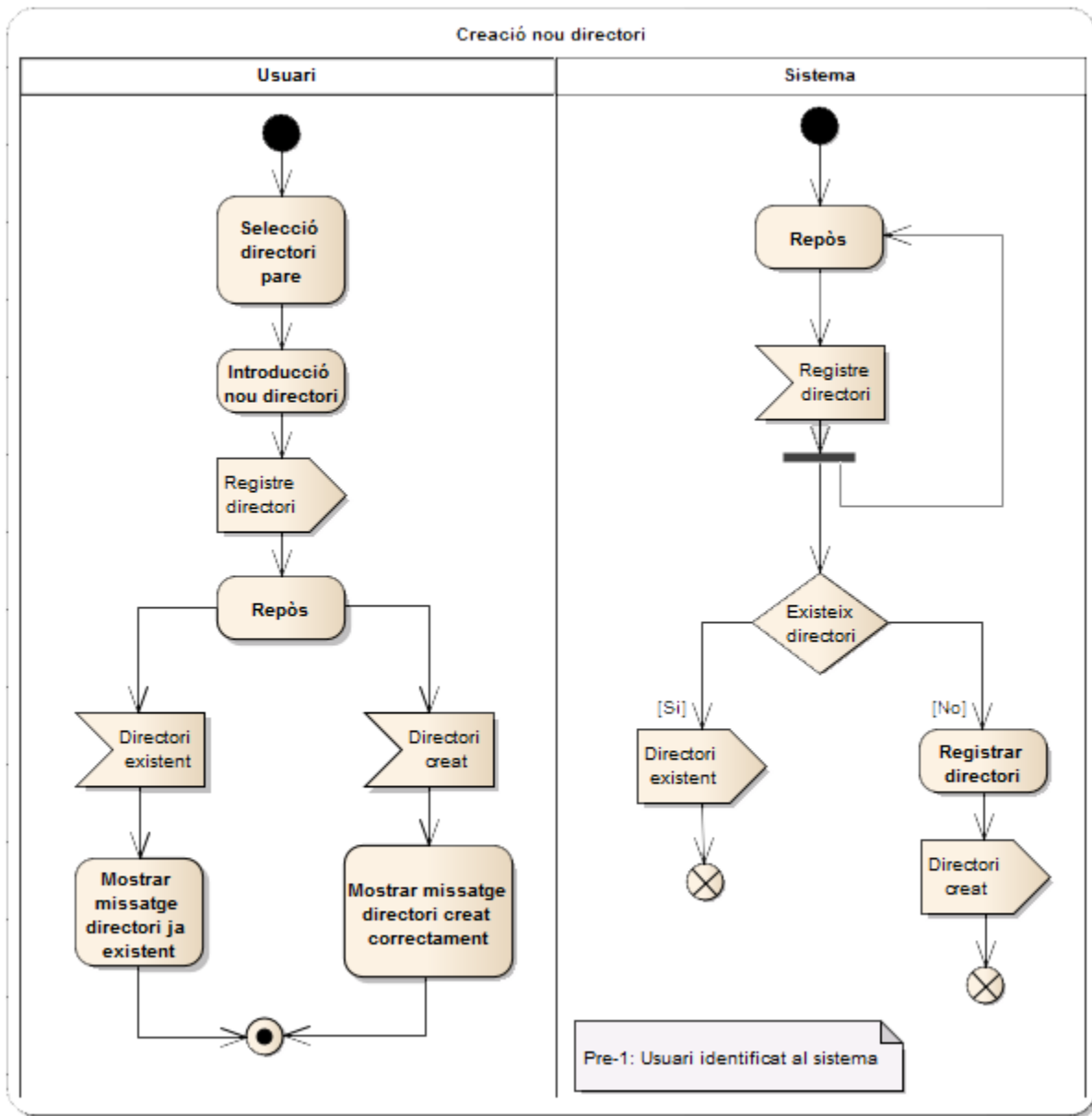
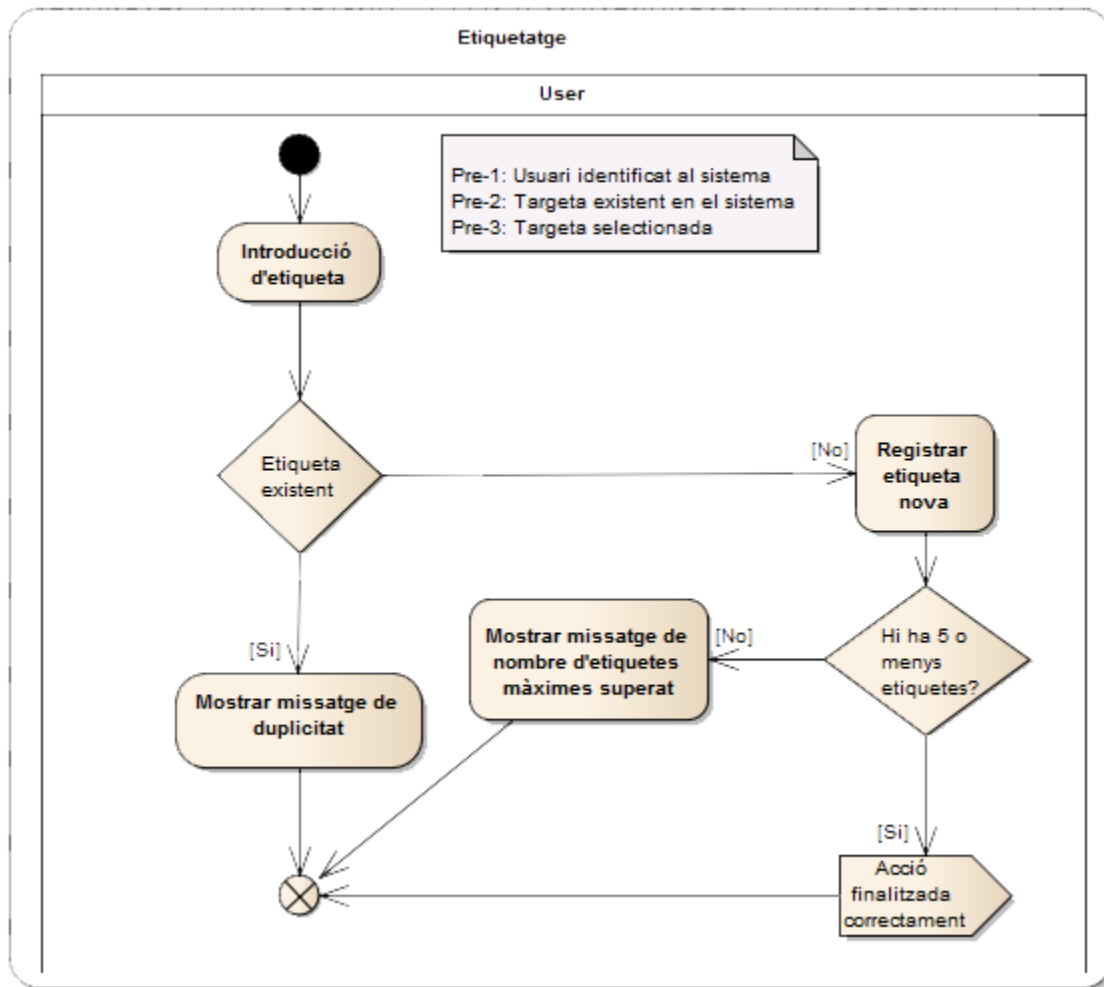


Diagrama 4: Creació nou directori

Alhora de crear un nou directori, l'usuari realitzarà la petició al sistema el qual comprovarà que no existeixi ja el directori a crear. En cas positiu el sistema registrarà el nou directori i informarà a l'usuari de la creació. En cas contrari s'informarà que el directori a crear ja existeix.

## Etiquetatge de targetes



L'usuari introdueix noves etiquetes a una targeta. Donat que el màxim d'etiquetes permès és 5, només registrarà en cas d'existir 5 o menys targetes.

En cas d'error o duplictat el sistema informará que no s'han registrat els canvis.

## Importació de nova targeta via sistema

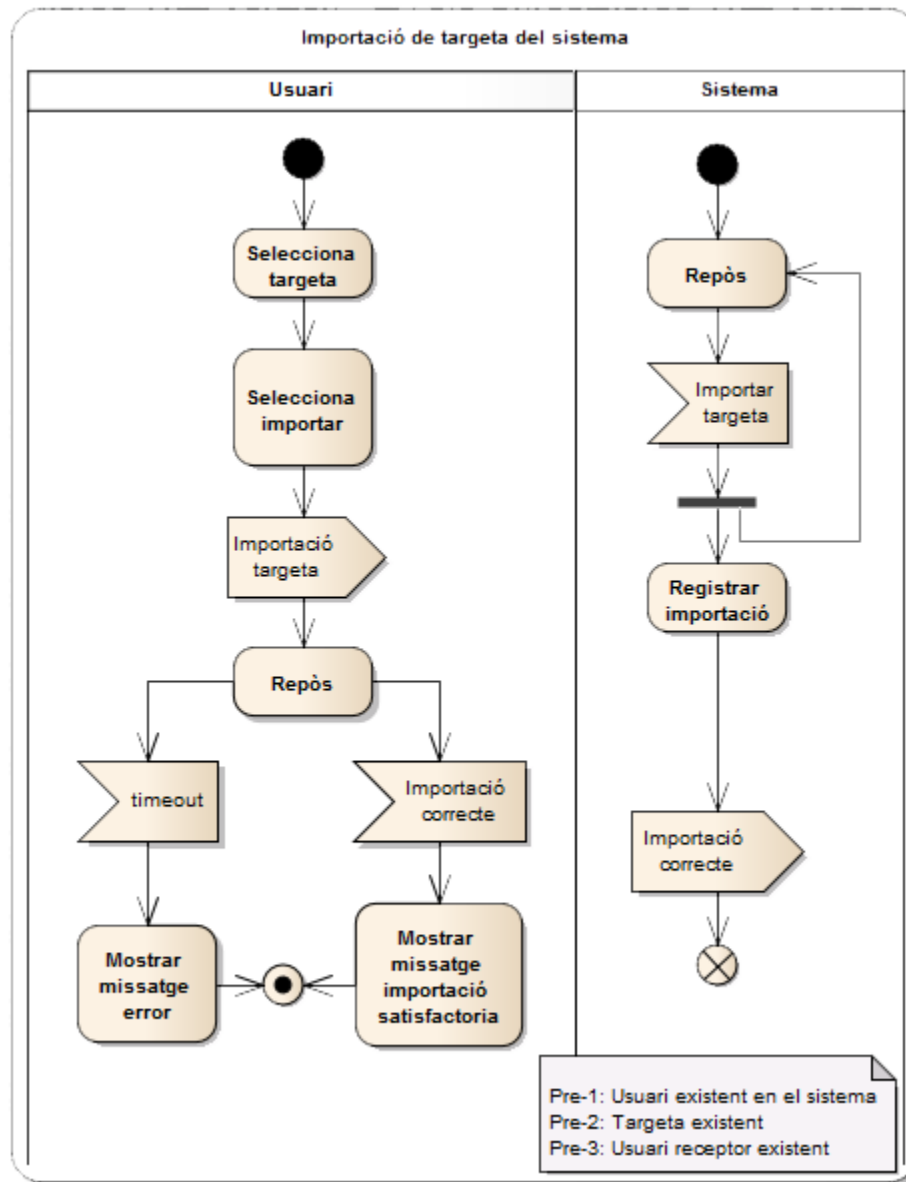


Diagrama 6: Importació de targeta

Un cop seleccionada l'opció d'importació d'una targeta, el sistema registrarà la importació copiant-la al directori de l'usuari.

Un cop acabada l'acció informarà que s'ha realitzar satisfactòriament.

## 4.7 Restriccions

A continuació es detallen les restriccions del sistema.

<b>Descripció:</b> El sistema ha de ser compatible per Android 4.x.
<b>Justificació:</b> Degut a la gran quantitat de terminals i versions Android existents al mercat, s'opta per desenvolupar per mòbils amb Android 4.x.
<b>Criteri de satisfacció:</b> Assegurar un mínim de funcionalitats natives i arribar alhora a un <b>~72%</b> dels terminals Android.

<b>Descripció:</b> El sistema ha de ser compatible amb tots els navegadors principals
<b>Justificació:</b> El sistema ha de poder ser utilitzat mitjançant web sense navegadors o complements especials.
<b>Criteri de satisfacció:</b> Sistema compatible amb Internet Explorer, Google Chrome i Mozilla Firefox.



## 4.8 Casos d'ús de producte

A continuació es presenta el diagrama de context amb els diversos casos d'ús de producte que conformaran el sistema:

### 4.8.1 Diagrama de context

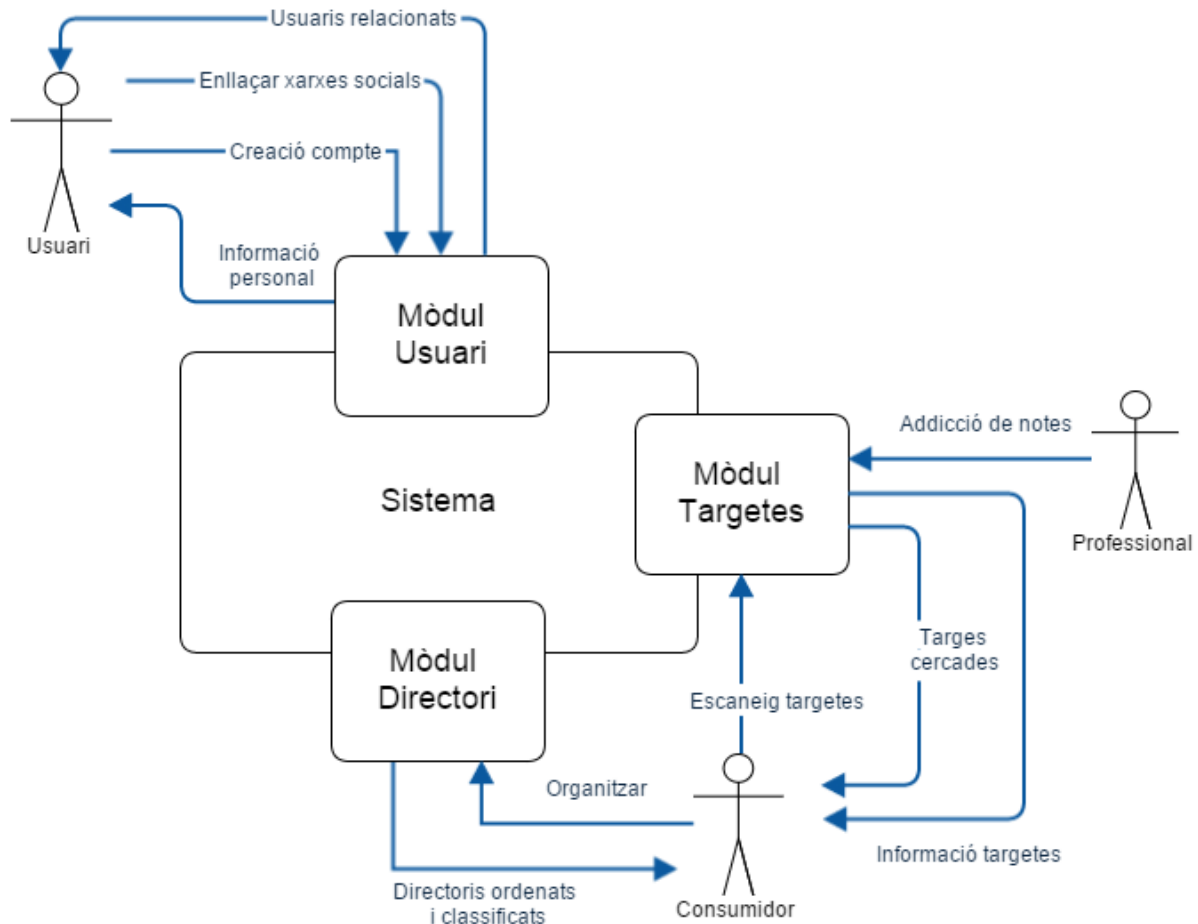


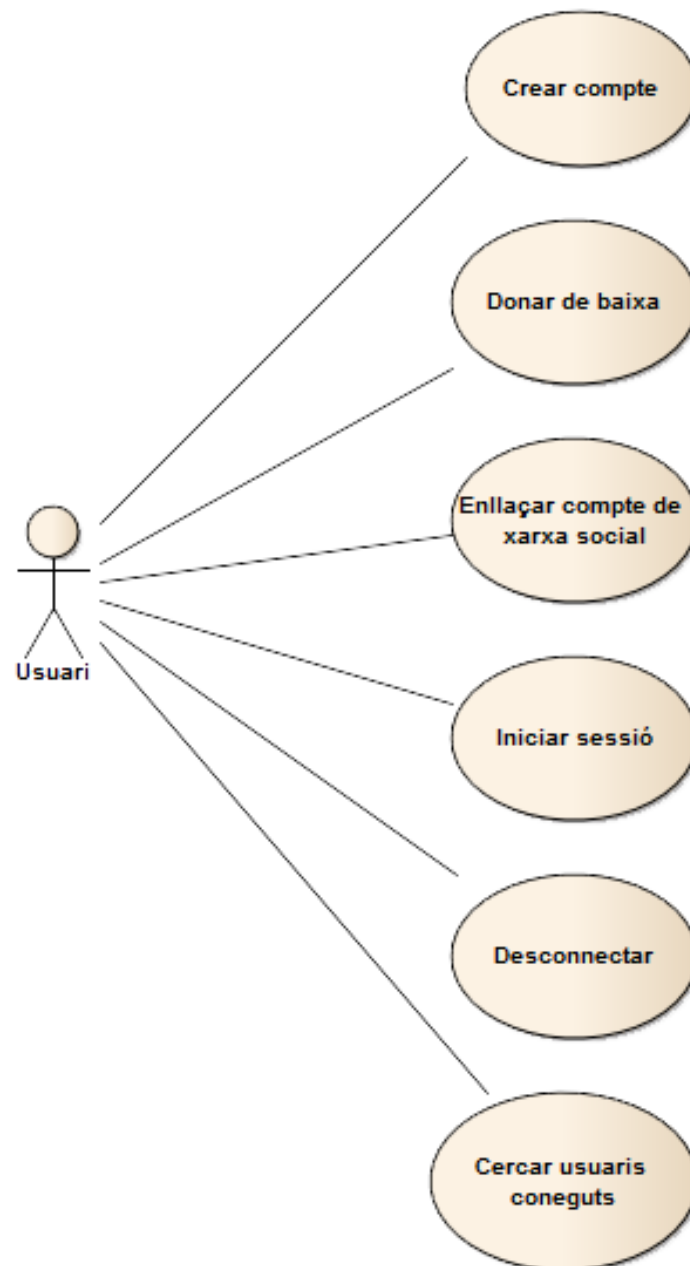
Diagrama 7: Diagrama de context

S'ha optat per dividir el sistema en tres mòduls, segmentant així les funcionalitats, obtenint com a resultat una millor integració i flexibilitat, alhora que s'obté una programació més llegible i manejable.

### 4.8.2 Limitacions de producte

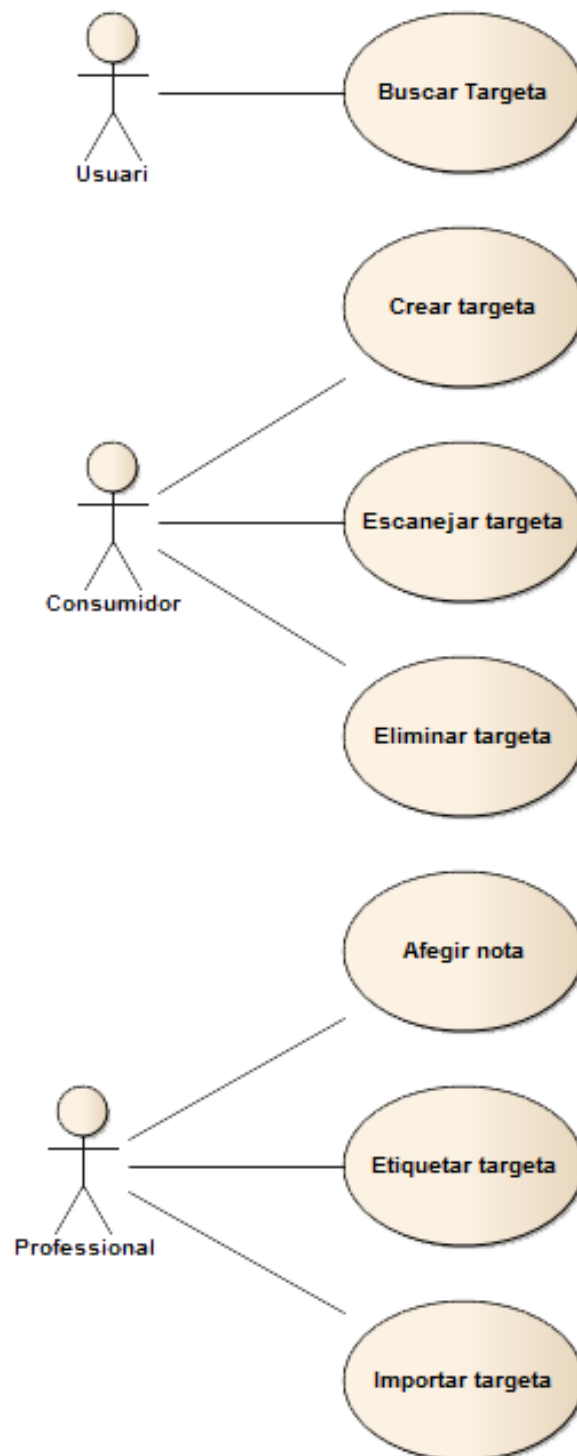
En aquesta secció s'exposen el conjunt de casos d'ús del sistema segmentats pels mòduls exposats al *diagrama 7* i per cada actor del sistema.

#### Gestió d'usuari



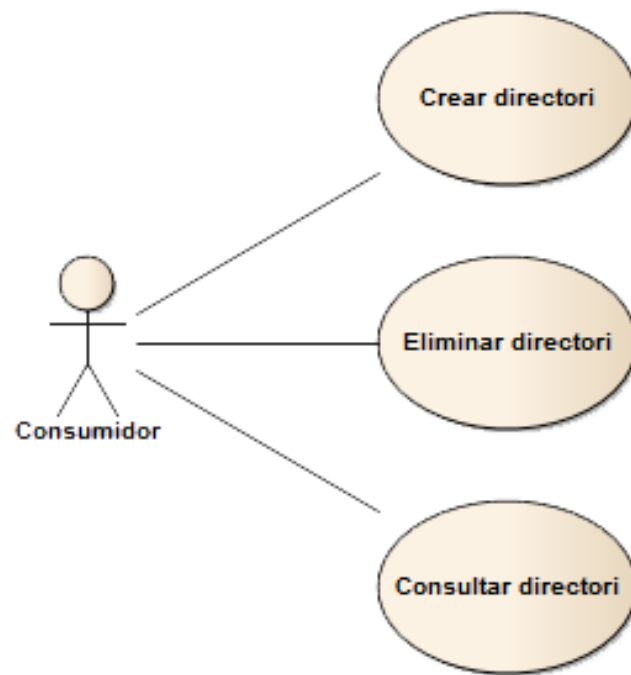
Imatge 2: Mòdul usuari

## Gestió de targetes



Imatge 3: Mòdul targeta

## Gestió dels directoris



Imatge 4: Mòdul directori

### 4.8.3 Llista de casos d'ús del producte

A continuació es presenten el conjunt d'interaccions entre els actors i el sistema, amb una breu descripció de la necessitat a satisfer.

#### Gestió d'usuari

- Crear compte: Registrar-se en el sistema com a usuari.
- Donar compte de baixa: Eliminar el compte d'usuari del sistema.
- Enllaçar compte de xarxa social: Addició de compte de xarxa social per cercar el conjunt d'usuaris als que pot tenir accés en el sistema.

#### Gestió de targetes

- Crear targeta: Introduir una nova targeta de contacte al sistema.
- Escanejar targeta: Processar informació d'una targeta de contacte física mitjançant OCR.
- Importar targeta: Guardar targeta del sistema d'un altre usuari en el propi directori.
- Etiquetar targeta: Assignar un tag a una targeta
- Eliminar targeta: Eliminar una targeta del sistema.
- Afegir nota: Afegir una anotació a una targeta.
- Cercar targeta: Cercar en el sistema un conjunt de targetes donats uns filtres.

#### Gestió directoris

- Crear directori: Crear nou directori de targetes al sistema.
- Eliminar directori: Eliminar directori del sistema.
- Modificar directori: Canviar el nom del directori.

## 4.8.4 Casos d'ús individuals

### 4.8.4.1 Gestió d'usuari

<b>Cas d'ús:</b> Crear compte d'usuari	
<b>Actor principal:</b> Usuari <b>Precondició:</b> - <b>Activació:</b> Un usuari decideix obrir un nou compte en el sistema. <b>Escenari principal:</b> <ol style="list-style-type: none"><li>1 L'usuari accedeix a la web o aplicació mòbil i pitja l'opció de crear nou compte.</li><li>2 El sistema reenvia a facebook per realitzar la validació de credencials</li><li>3 L'usuari accepta les condicions d'ús de la pàgina.</li><li>4 El sistema recull la validació de credencials per part de facebook i registra el nou compte.<ol style="list-style-type: none"><li>a. En cas d'existir l'usuari, es retorna els credencials pertinents i es mostra la pantalla de contingut.</li><li>b. En cas de no acceptar les condicions d'ús de la pàgina el sistema retorna a la pàgina inicial de la web o aplicació mòbil.</li></ol></li></ol>	
<b>Escenari de validació:</b> L'usuari s'identifica correctament en el sistema i se li dona accés a les diverses funcionalitats internes.	
<b>Satisfacció del client:</b> [ 5 ]	<b>Insatisfacció del client:</b> [ 5 ]

<b>Cas d'ús:</b> Donar compte de baixa	
<b>Actor principal:</b> Usuari <b>Precondició:</b> Usuari identificat en el sistema. <b>Activació:</b> Un usuari opta per donar de baixar el compte del sistema. <b>Escenari principal:</b> <ol style="list-style-type: none"><li>1 L'usuari selecciona dins de la secció de gestió d'usuari l'opció d'eliminar el compte.</li><li>2 El sistema sol·licita la confirmació de l'usuari.</li><li>3 L'usuari accepta l'eliminació del compte.</li><li>4 El sistema dona de baixa el compte, eliminant tota la informació referent a l'usuari.</li></ol>	
<b>Escenari de validació:</b> L'usuari no pot tornar a identificar-se en el sistema. El sistema li retorna que l'usuari no existeix.	
<b>Satisfacció del client:</b> [ 1 ]	<b>Insatisfacció del client:</b> [ 4 ]

#### 4.8.4.2 Gestió de targetes

[illegible]

<b>Cas d'ús:</b> Escanejar targeta	
<b>Actor principal:</b> Usuari	
<b>Precondició:</b> Usuari identificat en el sistema; Dispositiu amb càmera.	
<b>Activació:</b> L'usuari decideix introduir una targeta al sistema mitjançant l'escaneig.	
<b>Escenari principal:</b> <ul style="list-style-type: none"><li>1 L'usuari selecciona l'opció d'escanejar targeta.</li><li>2 El sistema activa la càmera.</li><li>3 L'usuari captura una imatge de la targeta.</li><li>4 El sistema valida els camps i dona l'opció d'editar-los.</li><li>5 L'usuari modifica els camps adients.</li><li>6 El sistema valida els camps i registra la nova targeta al sistema.<ul style="list-style-type: none"><li>a. Si algun camp és erroni, el sistema mostra missatge d'error i torna al pas 5.</li></ul></li></ul>	
<b>Escenari de validació:</b> L'usuari consulta el llistat de targetes disponibles on s'hi troba la nova targeta escanejada.	
Satisfacció del client: [ 3 ]	Insatisfacció del client: [ 3 ]







#### 4.8.4.3 Gestió dels directors

[illegible]

<b>Cas d'ús:</b> Canviar nom
<p><b>Actor principal:</b> Usuari</p> <p><b>Precondició:</b> Usuari identificat; Directori existent; Directori pertanyent a l'usuari.</p> <p><b>Activació:</b> Un usuari opta per canviar la visibilitat d'un directori.</p> <p><b>Escenari principal:</b></p> <ol style="list-style-type: none"> <li>1 L'usuari selecciona canviar nom del directori.</li> <li>2 El sistema mostra el nom actual</li> <li>3 L'usuari edita el nom del directori</li> <li>4 El sistema registra l'operació.</li> </ol>
<p><b>Escenari de validació:</b> El nom del directori és actualitzat a l'actual</p> <p>Satisfacció del client: [ 3 ]                      Insatisfacció del client: [ 4 ]</p>

## 5 Disseny

En aquesta secció es descriu el funcionament intern de l'aplicació i els motius que han comportat al disseny final. S'inicia amb la descripció de la tecnologia i arquitectura de l'aplicació i finalitza amb l'especificació de les capes de presentació, domini i dades del sistema.

### 5.1 Tecnologia

L'aspecte tecnològic del sistema és pot fragmentar en tres parts:

**Servidor** Implementat íntegrament en *Javascript*, donat que s'ha utilitzat *NodeJs* (veure secció 6.1.6) com a plataforma d'aplicació. La interacció amb BDD també s'ha realitzat en Javascript.

**Pàgina web** Realitzada en *Javascript*, *HTML* i *Css*. Com a framework per facilitar la implementació i agilitzar la interacció s'ha utilitzat *AngularJs* donat que respecte el patró model-vista-controlador (MVC).

**Aplicació Android** Desenvolupada sobre *Java*.

Així doncs, les tecnologies utilitzades són Javascript, HTML, Css, AngularJs, NodeJs i Java. A continuació es detalla cadascuna.



Imatge 5: Logotips html, js i css

### 5.1.1 HTML5

Utilitzat en la part de navegador del client, HTML5 (Hyper Text Markup Language) és la cinquena gran revisió del llenguatge bàsic de la World Wide Web.

Incorpora noves etiquetes com <nav> (bloc de navegació del lloc web) i <footer>, utilitzat pels peus de pàgina.

### 5.1.2 CSS

CSS (Cascading Style Sheets) és un llenguatge utilitzat per definir el conjunt d'estils (aspecte i format) dels elements d'HTML.

La seva principal funcionalitat és separar el disseny de la presentació del contingut propi. Fent més llegible la part de contingut i més net la definició de colors, disposició i fonts.

S'utilitza en la part del navegador del client i sol ser combinat amb uns estils predeterminats, en aquest cas, *Bootstrap* (*framework orientat al responsive design*), amb els establerts per requeriments del disseny o de l'aplicació desenvolupada.

### 5.1.3 JavaScript

JavaScript és un llenguatge de programació interpretat basat en el concepte de prototipatge (herència per delegació). És conegut sobretot pel seu ús en pàgines web, però també pot ésser utilitzat en altres aplicacions com servidors d'aplicacions (*NodeJs*) o en aplicacions mòbil mitjançant frameworks com *PhoneGap*.

Les relacions d'herència entre objectes no són per classes i subclasses, sinó individuals entre un objecte i un altre, anomenat el seu prototipus, al qual es deleguen les operacions i propietats que el primer desconeix.

Es tracta d'un llenguatge de tipatge dinàmic, el tipus està associat al valor i no a la variable, és a dir, una variable x en un moment donat pot ser un número i en un altre moment una cadena de caràcters.

A més, està format quasi en la seva totalitat per objectes, els quals són llistes associatives millorades amb la inclusió de prototipus. Les propietats i els valors poden ésser creats, canviats o eliminats en temps d'execució.

### 5.1.4 AngularJs



AngularJs és un framework de JavaScript de codi obert mantingut per Google. L'objectiu és augmentar el nombre d'aplicacions basades en el navegador amb capacitat del patró model-vista-controlador (MVC).

El framework adapta i amplia el HTML tradicional per servir contingut dinàmic mitjançant un data-binding bidireccional que permet la sincronització automàtica dels models i de les vistes.

Els objectius del disseny són:

- Dissociar la manipulació del DOM de la lògica de l'aplicació. Com a conseqüència millora la capacitat de testeig del codi.
- Dissociar la part del client d'una aplicació de la part del servidor, permetent que el treball de desenvolupament avanci en paral·lel.

Com s'ha comentat anteriorment, AngularJs segueix el patró MVC. Amb l'ús de la injecció de dependències, Angular porta els serveis tradicionals del costat del servidor, tals com controladors dependents de vistes, a les aplicacions web del client. En conseqüència, gran part de la càrrega de backend es redueix.

Un dels components més utilitzats són les directives, les quals permeten reutilitzar elements HTML i atributs que defineixen el data binding. Les més utilitzades en l'aplicació són:

**\$scope:** model d'informació. Conté les dades utilitzades en l'interacció de l'usuari i la lògica de negoci.

**ng-repeat:** Instància un element per cada objecte en la col·lecció. En l'exemple donat, tenim una llista de coses a fer amb una casella de confirmació. Només amb el codi podem representar tota la llista, sense necessitat de saber el nombre d'elements o d'afegir feina extra.

```
1. <ul class="unstyled">
2.   <li ng-repeat="todo in todos">
3.     <input type="checkbox" ng-model="todo.done">
4.     <span class="done-{{todo.done}}">{{todo.text}}</span>
5.   </li>
6. </ul>
```

**ng-controller:** Controladora que avalua les expressions d'HTML. Connecta la vista amb el model.

```
<div ng-controller="BeerCounter">
  <div ng-repeat="beerCount in beers">
    <ng-pluralize count="beerCount" when="beerForms"></ng-pluralize>
  </div>
</div>
```

Javascript, controladora per mostrar el plural de les cerveses:

```
angular.module('app', ['components'])

.controller('BeerCounter', function($scope, $locale) {
  $scope.beers = [0, 1, 2, 3, 4, 5, 6];
  if ($locale.id == 'en-us') {
    $scope.beerForms = {
      0: 'no beers',
      one: '{} beer',
      other: '{} beers'
    };
  } else {
    $scope.beerForms = {
      0: 'žiadne pivo',
      one: '{} pivo',
      few: '{} pivá',
      other: '{} piv'
    };
  }
});
```

Resultat de l'exemple de pluralitzar un comptador de cerveses:

```
no beers
1 beer
2 beers
3 beers
4 beers
5 beers
6 beers
```

### 5.1.5 Java



Java és un llenguatge interpretat, per tant necessita d'un programa intèrpret per a ser executat, en aquest cas, *Java Virtual Machine* (JVM). Això permet que sigui un llenguatge multiplataforma i sigui l'interpret l'específic a cada plataforma.

Les característiques principals són que és un llenguatge senzill, sense herència múltiple i amb l'abstracció dels punters (tot són punters, però a nivell de programació no poden ser accedits al igual que el C++).

Està orientat a objectes donat que es centralitza en la manipulació, creació i construcció d'objectes.

Un punt fort a destacar és el gran nombre de llibreries disponibles per al desenvolupament d'aplicacions, així com la gran comunitat de programadors que hi ha al darrere.

### 5.1.6 NodeJs



NodeJs és un entorn de programació en la capa de servidor dissenyat per escriure ràpidament aplicacions d'internet escalables. Arquitectura orientada a esdeveniments, asincrona amb I/O de dades. Aconsegueix que sigui lleuger i eficient, ideal per aplicacions en temps real que s'executen en dispositius distribuïts.

Està escrit sobre el llenguatge de programació JavaScript. La posada en marxa inicial és tant simple com el codi següent que ens permet tenir un servidor web escrit en Node que ens retorna "Hello World" en cada crida.

```
var http = require('http');

http.createServer(function (request, response) {
  response.writeHead(200, {'Content-Type': 'text/plain'});
  response.end('Hello World\n');
}).listen(8000);
```



## 5.2 Arquitectura

Per al desenvolupament de l'aplicació, s'ha decidit utilitzar l'arquitectura Model Vista Controlador (MVC). Es tracta d'un patró que separa les dades de l'aplicació, les interfícies d'usuari i la lògica de negoci en tres components diferents:

**Model:** És la representació de la informació amb la qual el sistema opera. Gestiona els accessos a la informació així com les actualitzacions. Envia a la vista aquesta part d'informació que en cada moment és sol·licitada.

**Controlador:** Respon als esdeveniments, normalment accions de l'usuari i invoca peticions al model en el cas de sol·licituds d'informació. També interactua amb la vista associada, és a dir, fa d'intermediari entre la vista (interaccions de l'usuari) i el Model (dades).

**Vista:** Presenta la informació en un format adequat per a la correcta interacció de l'usuari. Utilitza el controlador per gestionar les dades del sistema.

## 5.3 Capa de presentació

### Pàgina web

S'ha implementat amb AngularJs utilitzant les directives d'HTML per connectar amb la controladora i fer les crides necessàries per a actualitzar o obtenir informació sobre les targetes, directoris i usuaris.

També s'ha utilitzat el framework de bootstrap pels components de visualització.

### Android

Mitjançant els layout s'han definit les pantalles de l'aplicació i amb les activities les funcions corresponents a la interacció de l'usuari per donar resposta als esdeveniments entrants.

En el *diagrama 8* podem veure la interacció de les diverses tecnologies amb les tres capes.

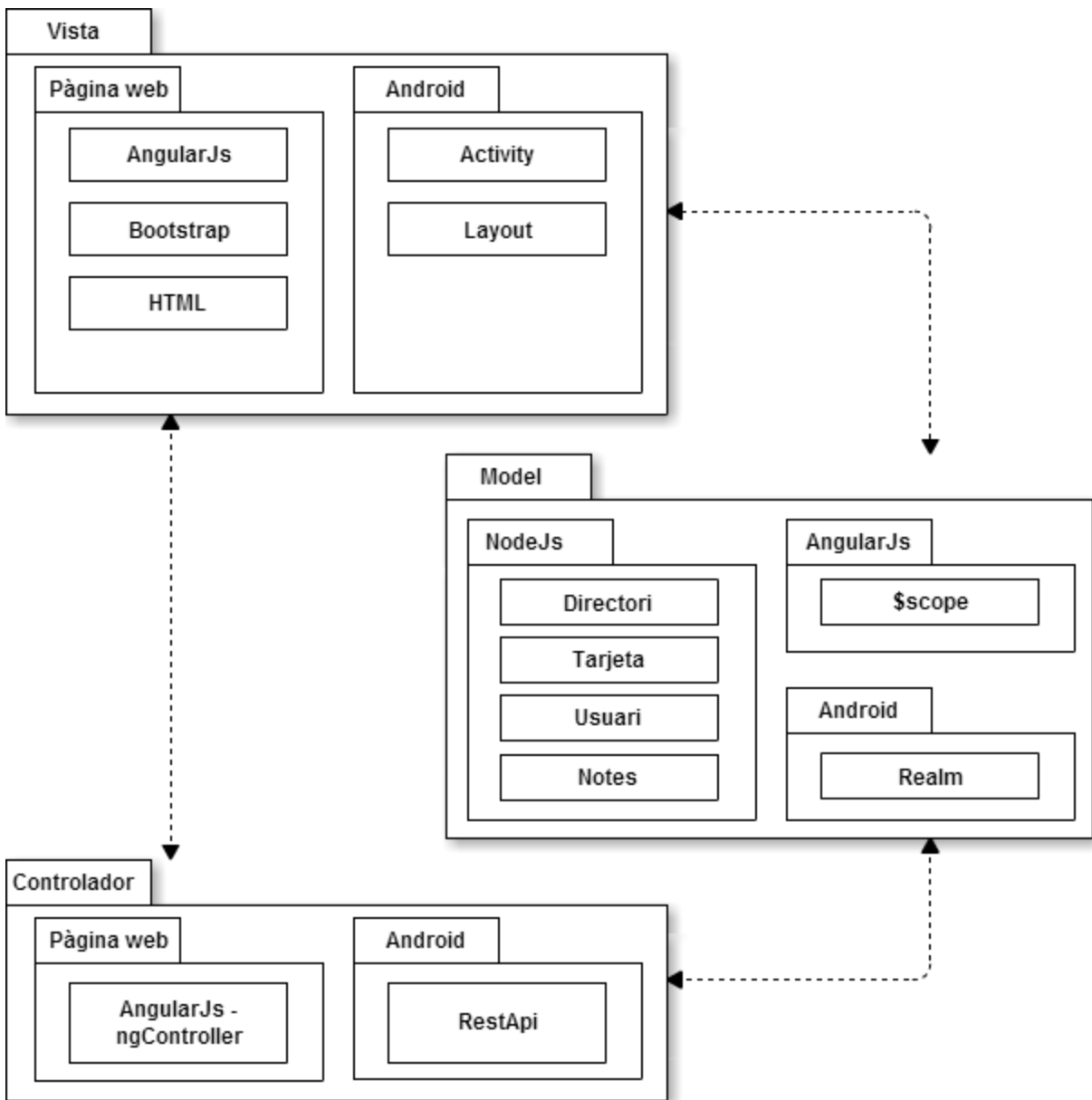


Diagrama 8: Capa de dades

## 5.4 Capa de domini

Aquesta capa conté la lògica del projecte i per tant és on es realitzen el conjunt de comprovacions intermitges de les dades, conté la transició de les pantalles i s'encarrega de comunicar-se entre la capa de dades i la de presentació.

### Pàgina web

La directiva ngController d'Angular ens permet desacoblar la lògica de negoci de la capa de presentació de la capa de domini. En les diverses controladores definim les accions a realitzar donat un conjunt d'interaccions de l'usuari. També s'encarrega de l'obtenció i modificació de les dades vers el servidor.

El conjunt de controladores utilitzades es troba detallat en la *secció 7.2.6*.

### Android

Conté les crides al servidor per obtenir i modificar la informació. També conté la lògica per cridar a la capa de dades i guardar la informació en local.

Mitjançant listeners, és capaç de respondre als esdeveniments causats de l'interacció de l'usuari amb l'aplicació i executar la lògica corresponent.

Les crides a servidor, així com el listeners utilitzats es troben explicats més endavant, en la *secció 7.2.9*.

## 5.5 Capa de dades

En aquesta capa s'hi gestiona tot el que està relacionat amb les dades del sistema. Es controla l'accés a la base de dades del servidor i de l'aplicació Android. També es realitzen les últimes comprovacions de modificació d'informació per validar que tot és correcte i no s'està guardant informació incorrecte que ha estat omesa per la capa de domini.

Aquesta última comprovació es sol fer en cas de que l'usuari modifiqui l'execució esperada en la capa de domini i es salti les comprovacions realitzades anteriorment.

Les bases de dades utilitzades són *MongoDB* en servidor i *Realm* en l'aplicació Android. L'especificació i interacció amb aquestes es troba detallada posteriorment en les seccions 7.1.4 i 7.2.10 respectivament.

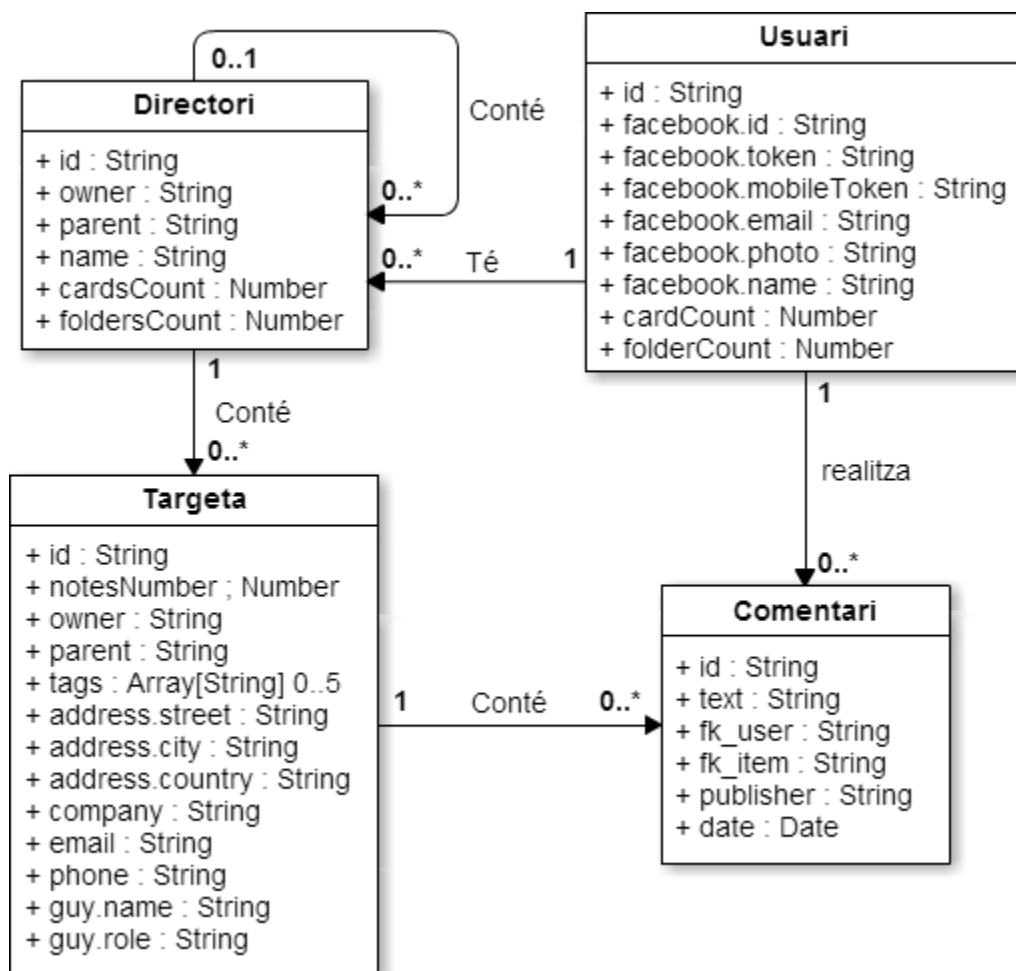


Diagrama 9: Base de dades

## 6 Implementació

En aquesta secció es descriu l'entorn de desenvolupament utilitzat en el Projecte, així com els detalls de la implementació dels diversos components que el conformen.

### 6.1 Entorn de desenvolupament

Tot el desenvolupament s'ha dut a terme utilitzant com a sistema operatiu Ubuntu 12.04.

#### 6.1.1 Git - Gestió de versions

Git és un programari de sistema de control de versions pensat en l'eficiència i confiabilitat de manteniment de versions d'aplicacions.



Primer de tot s'ha creat un projecte a bitbucket: [bitbucket.org/afranzi/pfc](https://bitbucket.org/afranzi/pfc)

```
# Instal·lació Git
sudo aptitude install git
# Generació de signatura per interactuar amb el compte de bitbucket
ssh-keygen -t rsa -C "franzicross@gmail.com"
# Còpia de la clau pública generada a bitbucket
cat ~/.ssh/id_rsa.pub
```

Un cop copiada la clau pública ja podem realitzar commits al projecte.

#### 6.1.2 Sublime Text



Sublime Text és un editor de text orientat a la programació. Força útil en el camp de JavaScript i HTML.

Per a la instal·lació s'ha d'afegir el repositori de sublime al repositori d'aplicacions del sistema i tot seguit instal·lar-lo.

```
sudo add-apt-repository ppa:webupd8team/sublime-text-2
sudo apt-get update
sudo apt-get install sublime-text
```

### 6.1.3 NodeJs + NPM

Per programar la part de servidor és necessari la instal·lació de NodeJs.

```
sudo aptitude install nodejs
```

A més, s'ha utilitzat *NPM*, Node Packaged Modules, per gestionar el conjunt de llibreries utilitzades en el desenvolupament del servidor. NPM ens permet instal·lar diversos paquets de manera local, és a dir, dins la carpeta del projecte, o de manera global en el sistema, cosa que ens permet reutilitzar el paquet en diversos projectes a la vegada.

Una altra funcionalitat que ens serveix d'ajuda és la definició de l'arxiu ***package.json*** en el qual hi trobem els mòduls de node utilitzats i la versió d'aquests. Això ens permet poder portar l'aplicació a diferents sistemes sense necessitat de preocupar-nos de què necessitem, donat que l'instal·lació dels mòduls utilitzats es realitza de la següent manera:

```
npm install
```

El format de l'arxiu package.json és el següent:

```
{
  "name": "smart-wallet",
  "version": "0.0.1",
  "private": true,
  "dependencies": {
    "supervisor": "*",
  },
  "scripts": {
    "start": "./node_modules/.bin/supervisor app.js",
    "test": "./node_modules/.bin/mocha -u tdd -R spec"
  }
}
```

En l'arxiu també podem definir els scripts que podran ésser executats amb npm.

- **npm start** - Ens permet iniciar el servidor
- **npm test** - Executa els tests unitaris.

### 6.1.4 MongoDB

MongoDB és una base de dades, NoSQL, de documents open-source. Serà utilitzada per emmagatzemar el conjunt de targetes i directoris creats pels usuaris. També contindrà la informació dels usuaris així com els tokens de credencials.



```
# Importació de la clau pública utilitzada pel gestor de paquets
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
# Creació del fitxer de MongoDB
echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | sudo tee
/etc/apt/sources.list.d/mongodb.list
# Actualització del repositori
sudo apt-get update
# Instal·lació dels paquets de MongoDB
sudo apt-get install -y mongodb-org
```

Un cop això ja tenim instal·lat MongoDB en el sistema i ja podem començar a interactuar amb la BDD.

No obstant, per facilitar la lectura i testeig durant el desenvolupament s'ha utilitzat RoboMongo, una aplicació client per gestionar la BDD.

```
wget http://robomongo.org/files/linux/robomongo-0.8.4-x86_64.deb -O robomongo.deb
sudo dpkg -i robomongo.deb
```

### 6.1.5 Eclipse ADT

El desenvolupament de l'aplicació Android s'ha dut a terme mitjançant Eclipse ADT (*Android Development Tools*). La pàgina d'Android ja ens ofereix un paquet d'Android amb tot preparat:

- Eclipse + ADT plugin
- Android SDK tools
- Android Platform-tools
- Emulador d'Android.

```
# Ens descarreguem el paquet d'android.  
wget https://dl.google.com/android/adt/adt-bundle-windows-x86_64-20140702.zip \  
-O eclipseADT.zip  
unzip eclipseADT.zip
```

Dins de la carpeta creada d'eclipseADT tenim l'IDE d'eclipse i les eines de desenvolupament d'Android.

### 6.1.6 Nginx

*nginx* és un servidor web / servidor intermediari (servidor proxy) lleuger d'alt rendiment. És programari lliure i de codi obert.

Donat que el sistema consta d'un servidor web, ens és necessari l'accés al servidor des de l'aplicació mòbil. Així doncs, s'ha prosseguit a instal·lar nginx per publicar la web a la wifi interna de casa.



```
sudo apt-get update  
sudo apt-get install nginx  
# nginx s'inicia automàticament un cop instal·lat. Per saber la ip a on fer les  
# peticions executem la següent comanda:  
ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\/.*$//'
```

Un cop coneguda l'adreça ip podem accedir a nginx des de qualsevol dispositiu connectat a la xarxa local, i veure la següent pàgina:



# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

Imatge 6: Missatge per defecte de nginx

Donat que s'ha configurat el servidor de NodeJs al port 3000, s'haurà de configurar correctament nginx perquè al introduir la ip ens redirigeixi a la pàgina de la nostra aplicació.

```
#/etc/nginx/conf.d/wallywallet.com.conf
server {
    listen 80;
    server_name 192.168.1.10;

    location / {
        include proxy_params;
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Accept-Encoding "";
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Podem veure com per la ip `192.168.1.10` li demanem a nginx que redirigeixi la crida internament a `http://localhost:3000`

## 6.2 Detalls de la implementació

En aquest apartat es prossegueix a detallar els punts més importants de cada fragment que conforma el sistema.

### 6.2.1 NodeJs

El funcionament del servidor és detallat en l'arxiu *app.js* el qual és el primer en ésser cridat al moment d'iniciar el servei amb `npm start`.

Es configuren el port del servei, així com la base de dades a utilitzar i diversos paràmetres de configuració i rutes de pàgines web.

```
var express = require('express');
var app     = express();
var port    = process.env.PORT || 3000;
var mongoose = require('mongoose');
var flash   = require('connect-flash');
var configDB = require('./config/config.js'); //DBB properties
var passport = require('passport');
```

En la figura x podem veure el conjunt de mòduls a utilitzar en l'inici. *Mongoose* i *Passport* seran explicats en les següents seccions (7.2.3 i 7.2.4).

*Express* és un framework minimalista que corre per sobre de NodeJs que facilita la posada en marxa. Per altre banda *flash* és un mòdul que ens permetrà passar informació entre les diverses vistes de la pàgina web.

En el següent codi, es defineix la connexió a base de dades. En aquest cas la informació es troba a l'arxiu de *config.js* que s'ha importat anteriorment.

```
mongoose.connect(configDB.db.prod); // connect to our production database
// configDB.db.prod = "mongodb://localhost:27017/mongoose-wallet"
```

La configuració de com NodeJs ha de respondre a les crides externes es troba dins del *app.configure*.

```
// Configuration
app.configure(function(){
  app.use(express.logger('dev')); // log every request to the console
  app.use(express.cookieParser()); // read cookies (needed for auth)
  app.use(express.bodyParser()); // get information from html forms

  app.set('views', __dirname + '/views');
  app.set('view engine', 'ejs');
```

```
app.use(express.session({ secret: 'SecretApocalypseNow Andromeda' }));

//include the Passport middleware
app.use(passport.initialize());
app.use(passport.session()); // persistent login sessions
app.use(flash()); // use connect-flash for flash messages stored in session

app.use(require('stylus').middleware({ src: __dirname + '/public' }));
app.use(express.methodOverride());
app.use(express.static(__dirname + '/public'));
app.use(app.router);
});
```

*“app.use(require('stylus').middleware({ src: \_\_dirname + '/public' }));”* indica que des de l’exterior la única part que serà accessible serà tot el que estigui dins del directori public. Així podem tenir les configuracions i lògica de backend privada sense necessitat de que l’usuari pugui veure el codi ni de preocupar-nos de que s’hi intenta accedir.

Per altre banda, el conjunt de peticions disponibles estan definides en l’arxiu routes.js que és importat de la següent manera:

```
// load our routes and pass in our app and fully configured passport
var routes = require('./routes/routes.js')(app, passport);
```

En l'estructura de directoris del codi trobem la configuració amb els arxius pertinents a base de dades i credencials d'identificació. A *models* tenim la definició dels documents de MongoDB; *node\_modules* conté els mòduls especificats en el *package.json*. La carpeta de *views* conté els arxius HTML que seran renderitzats en la part de client.



## 6.2.2 Forever

Forever és un modul global (instal·lat en el sistema) de NodeJs que permet el control del servidor i s'assegura que sempre està en funcionament.

```
# Instal·lació  
sudo npm install -g forever
```

Un cop instal·lat només ens hem d'assegurar d'iniciar el servidor utilitzant forever, per a això utilitzarem un script que pot ésser llançat manualment o cronejat a l'iniciar-se la màquina per si es dona el cas de que la màquina cau, s'iniciï automàticament.

```
#!/bin/bash  
# Check that process doesn't exists  
if [ $(ps -e -o uid,cmd | grep $UID | grep node | grep -v grep | wc -l | tr -s "\n") -eq 0 ]  
then  
    echo "Starting Server!"  
    export PATH=/usr/local/bin:$PATH  
    forever start --sourceDir /root/Homeland/Web app.js >> /root/Homeland/logs/log.txt 2>&1  
fi
```

En l'arxiu de crontab tenim definit la crida del script amb l'inici de la màquina:

```
@reboot /root/starter.sh
```

### 6.2.3 Passport + Facebook

Passport és un middleware d'autenticació per NodeJs. Permet l'autenticació mitjançant OAuth 2.0.

Donat que un dels requisits era la interacció amb les xarxes socials s'ha implementat Facebook com a proveïdor.

Per al seu funcionament és necessari importar el mòdul de passport-facebook i obtenir un identificador d'aplicació de facebook i el secret corresponent. Per a això haurem d'anar a la secció de desenvolupadors de facebook i registrar la nostre aplicació.

```
var passport = require('passport')
    , FacebookStrategy = require('passport-facebook').Strategy;

// expose this function to our app using module.exports
module.exports = function(passport) {
  passport.use(new FacebookStrategy({
    clientID      : configAuth.facebookAuth.clientID,
    clientSecret  : configAuth.facebookAuth.clientSecret,
    callbackURL   : configAuth.facebookAuth.callbackURL
  }),
  // facebook will send back the token and profile
  function(token, refreshToken, profile, done) {
    var fb = new FB({
      appId: configAuth.facebookAuth.clientID,
      secret: configAuth.facebookAuth.clientSecret
    }).setAccessToken(token);

    // asynchronous
    process.nextTick(function() {
      // find the user in the database based on their facebook id
      User.findOne({'facebook.id' : profile.id}, function(err, user) {
        // if there is an error, stop everything and return that
        if (err) return done(err);
        else login(fb, user, token, refreshToken, profile, false, done);
      });
    });
  });
});
```

Quan un usuari s'intenti identificar a la nostra aplicació serà redirigit a facebook per demanar que accepti les condicions d'ús i en cas d'acceptar-les serà redirigit a la nostre funció d'identificació on podem registrar l'usuari o buscar-lo en cas d'existir prèviament.

Passport retornarà un credencial d'identificació que serà guardat en les cookies. En cada petició que es faci a servidor serà comprovat. En cas de ser invàlid serà retornat a la pàgina d'inici.

## 6.2.4 Mongoose

Mongoose és un mòdul d'ORM (Object-relational Mapping) que ens permet interactuar de manera fàcil amb MongoDB utilitzant modelatge d'objectes.

Mitjançant Mongoose definirem els esquemes dels objectes de Targetes, Directoris, Usuaris i Comentaris. Cada esquema en un fitxer separat.

```
var mongoose = require('mongoose'),
    Schema = mongoose.Schema;

var UserSchema = new Schema({
  facebook : {
    id      : { type: String, required: true, index: { unique: true }},
    token   : String,
    mobileToken : String,
    email   : String,
    photo   : String,
    name    : { type: String, required: true, index: { unique: false }},
  },
  cardCount : { type: Number, default: 0},
  folderCount : { type: Number, default: 0},
  dummy     : { type: Boolean, default: false},
});
```

En cada esquema serà necessari importar el mòdul. Com a tipatge accepta les Primitives bàsiques de String, Number, Boolean, Date i Array. S'hi poden definir valor per defecte i especificar quins camps són obligatoris.

A més, s'hi poden indicar quins camps MongoDB utilitzarà com a índex per agilitzar la posterior cerca. En el nostre cas s'ha utilitzat l'identificador d'usuari de facebook, donat que necessitarem consultar l'usuari força sovint mitjançant aquest id. També s'ha inclòs el nom d'usuari indicant que és un index no únic (múltiples usuaris poden tenir el mateix nom, però no mateix id).

Ara en tot document de backend on vulguem utilitzar l'esquema d'usuari només caldrà importar l'esquema.

Mongoose ens posa a disposició entre altres les següents funcions:

- `findOne(query)`: retorna el primer que concordi amb la query realitzada.
- `find(query)`: retorna el conjunt d'elements.
- `object.save()`: persisteix el document a base de dades.
- `object.remove()`: elimina el document de base de dades.

En el cas de tenir molts documents que concordin amb una mateixa condició, s'haurà de paginar la consulta.

```
var query = {'facebook.id': {$in : (friends)}, 'facebook.name': { $regex: search, $options: 'i'}};

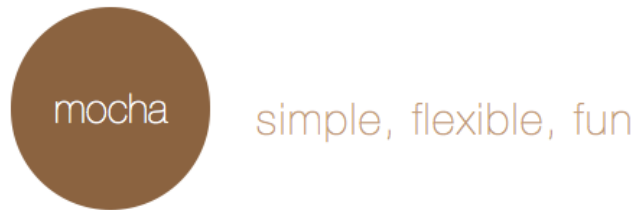
User.find(query, null, {skip: page*items, limit: items, sort: {'facebook.name': 1}},
  function(err, wallyFriends) {
    res.status(200).send(wallyFriends);
  }
);
```

En la figura x, es realitza una consulta dels usuaris que el seu identificador està inclòs dins del llistat *friends* i que contenen en el nom la paraula compresa en la variable *search*. *skip*, defineix el nombre d'elements a ignorar i *limit* el nombre d'elements a obtenir. *sort* ens retorna la cerca ordenada pel nom de l'usuari.



### 6.2.5 Mocha

Mocha és un framework de testeig de Javascript que s'executa sobre NodeJs i sobre la part de client i permet crides asíncrones.



Els tests es troben en el directori test i són executats mitjançant la comanda de la figura x. Això ens permet definir els tests en diversos arxius en funció de les funcionalitats a testear.

```
./node_modules/.bin/mocha -u tdd -R spec"
```

La programació dels tests consta de:

- **describe** Defineix el nom del test. Es poden anidar.
- **it** Test a executar
- **before(), after(), beforeEach(), afterEach()**: funcions a executar abans i després de cada test. Ens permet definir un comportament comú en un conjunt de tests.

```
describe('User', function(){
  describe('#save()', function(){
    it('should save without error', function(done){
      var user = new User('Luna');
      user.save(function(err){
        if (err) throw err;
        done();
      });
    })
  })
})
```

## 6.2.6 AngularJs

### Directive

Permeten estendre el comportament dels documents d'HTML. S'ha utilitzat per definir l'esdeveniment d'intro en els formularis.

```
app.directive('ngEnter', function () {
    return function (scope, element, attrs) {
        element.bind("keydown keypress", function (event) {
            if(event.which === 13) {
                scope.$apply(function (){
                    scope.$eval(attrs.ngEnter);
                });

                event.preventDefault();
            }
        });
    };
});
```

Ahora d'ésser utilitzada només cal incloure l'etiqueta ng-enter i definir quina funció executar de la controladora en la que es troba.

```
<form>
  <textarea ng-enter="createComment(comment)"
            class="form-control" ng-model="comment"
            maxlength="120 ">
  </textarea>
</form>
```

### ngController

ngController són funcions d'AngularJs que ens permeten definir una controladora en un entorn determinat. Aquestes controladores són utilitzades per contenir, modularitzar i reaprofitar el codi.

En la pàgina web s'han definit els següents controladors:

- CardListsCtrl: Conté la lògica de gestionar llistes de targetes. Així doncs permet modificar, consultar i eliminar targetes.
- CreateCardsCtrl: Controladora encarregada de registrar targetes i directoris al sistema.
- ProfileCtrl: Gestiona el perfil d'usuari. Mostra la informació i permet donar de baixa el compte del sistema.
- UserSearchCtrl: Controladora amb la lògica de gestió de cerques d'usuaris i targetes. També pagina els resultats obtinguts per realitzar crides òptimes que no carreguin la transacció ni l'ús de base de dades.

Cada controladora està definida en un arxiu per separat dins del directori public de documents JavaScript.

```
var app = angular.module('web', ['ui.bootstrap', 'ngRoute']);

function SearchCtrl($scope, $http, $location) {

    $scope.filter = null;
    $scope.items = [];
    $scope.userInfo = null;
    $scope.numItems = 0;

    $scope.init = function(filter) {
        if(!filter) {
            filter = "";
        }

        $scope.filter = decodeURI(filter);
        $scope.getUserResults($scope.filter);
        $scope.getUserInfo($scope.filter, 0, $scope.pageSizeSearch);
    }

    $scope.getUserInfo = function(filter, page, items) {
        $scope.getItems('/search/users', filter, page, items);
    }

    $scope.getCardsInfo = function(filter, page, items) {
        $scope.getItems('/search/cards', filter, page, items);
    }

    $scope.getTagsInfo = function(filter, page, items) {
        if(!filter) return;
        $scope.getItems('/search/tags', filter, page, items);
    }

    ...
}
```

El següent codi mostra com definir les controladores. La funció ng-init és cridada en la inicialització de cadascuna.

```
<body ng-controller="SearchCtrl" ng-init="init('<%= search %>')">
  <div ng-controller="CardsListsCtrl" ng-init="initSearch('<%= user.facebook.id %>')">
    ....
  </div>
</body>
```

### 6.2.7 restApi

S'ha definit un conjunt de crides que permetin l'accés i modificació de les dades del sistema. Les crides permeses són GET (obtenció d'informació) i POST (creació, modificació o eliminació d'informació).

NodeJs permet la definició de múltiples fixers de rutes. Però, s'ha de tenir en compte que quan un usuari realitza una crida NodeJs busca seqüencialment l'entrada de la funció corresponent i per tant s'ha de vigilar en no reescriure o no utilitzar wildcards en arxius intermitjos.

```
// routes
var userRoutes = require('./routes/restApi/restUser.js')(app, passport);
var restApi    = require('./routes/restApi.js')(app, passport);
var routes     = require('./routes/routes.js')(app, passport);
```

En totes les crides s'hi passa el mòdul Passport donat que és necessari estar identificat en el sistema per poder interactuar amb la informació interna.

```
/* Obté comentari */
app.get('/comment/:id', isLoggedIn, function (request, response) {
    var containerId = request.params.id;
    if(containerId != null && containerId != undefined) {
        Comment.findAllById(containerId, function (comments) {
            response.send(comments);
        });
    } else {
        response.send(403);
    }
});
```

La funció *isLoggedIn* comprova que l'usuari que ha realitzat la crida estigui identificat correctament, en cas contrari retornarà un 401. “:id” defineix el parametre identificador de la targeta que conté els comentaris a cercar.

La figura x, permet registrar un nou comentari. El contingut del comentari és passat en el cos de la petició.

```
/* Crea comentari */
app.post('/comment', isLoggedIn, function (request, response) {
    var commentJson = request.body;
    commentJson.fk_user = request.user.facebook.id;
    var newComment = new Comment(commentJson);
    newComment.save(function(err, comment) {
        if (err) {
            response.send(403);
        }
    });
});
```

```

        } else {
            response.send(comment).status(200);
        }
    });
});

```

### 6.2.8 Manifest.xml

Tota aplicació Android consta de l'arxiu Manifest.xml on es defineix el conjunt de pantalles que formen l'aplicació i les característiques d'aquestes.

Consta de tres principals parts:

Versió de la sdk d'Android. La versió 14 permet l'execució de l'aplicació en sistemes android 4.0 o superiors.

```

<uses-sdk
    android:minSdkVersion="14"
    android:targetSdkVersion="14" />

```

Permisos d'interacció amb el sistema que són sol·licitats a l'usuari en el moment de l'instal·lació i que defineixen el grup d'accions que l'aplicació pot realitzar sobre el dispositiu. Són necessaris l'accés a internet, la lectura i escriptura en la memòria externa del dispositiu.

```

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

```

Pantalles i les corresponents activitats que contenen la lògica d'execució.

```

<application android:largeHeap="true" >
    <activity
        android:name="com.facebook.LoginActivity"
        android:label="@string/app_name"
        android:theme="@android:style/Theme.Translucent.NoTitleBar" />
    ....

```

## 6.2.9 HttpClient

Les crides a servidor han de ser asíncrones donat que no es pot bloquejar el fil d'execució principal de l'aplicació. Per a la implementació d'aquesta funcionalitat s'utilitza la llibreria AsyncHttpClient.

Permet fer crides GET i POST i persistir les cookies d'identificació a l'aplicació. Així doncs, un cop identificats mitjançant Facebook, el servidor ens retorna la cookie i ja podem executar correctament crides a la restApi definida en la *secció 7.2.7*.

```
public class RestClient {
    private static final String BASE_URL = "http://www.wallywallet.net/";

    private AsyncHttpClient client = new AsyncHttpClient();
    private PersistentCookieStore cookieStore;

    private RestClient(Context context) {
        cookieStore = new PersistentCookieStore(context);
        client.setCookieStore(cookieStore);
    }

    public void get(String url, RequestParams params,
        AsyncHttpResponseHandler responseHandler) {
        client.get(getAbsoluteUrl(url), params, responseHandler);
    }

    public void post(String url, RequestParams params,
        AsyncHttpResponseHandler responseHandler) {
        client.setEnableRedirects(true);
        client.post(getAbsoluteUrl(url), params, responseHandler);
    }
}
```

En els handlers definim la lògica a executar un cop la petició http és resolta.

### 6.2.10 Realm

Realm és una base de dades NoSql orientada a mòbil. Al igual que Mongoose, *secció 7.2.4*, és *ORM* i permet el mapeig a models definits mitjançant classes amb només *setters* i *getters*.

És possible disposar de múltiples bases de dades per a una mateixa aplicació, cadascuna de les quals serà guardada en un fitxer diferent.

Cada cop que es desitja gestionar dades s'ha d'iniciar una instància i un cop realitzades les transaccions s'ha de finalitzar.

```
Realm realm = Realm.getInstance(this, type.name() + ".realm");
realm.beginTransaction();

/* Save items in realm to be accessible in offline mode */
if(Type.CARD.equals(type) && isOnline) {
    for(AbstractItem item: cards) {
        Card card = (Card) item;
        CardRealm itemRealm = realm.createObject(CardRealm.class);
        itemRealm = card.clone(itemRealm);
    }
} else if(Type.FOLDER.equals(type) && isOnline) {
    for(AbstractItem item: cards) {
        Folder folder = (Folder) item;
        FolderRealm itemRealm = realm.createObject(FolderRealm.class);
        itemRealm = folder.clone(itemRealm);
    }
}

realm.commitTransaction(); //discardTransaction reverse all changes
```

La figura x, persisteix a Realm els llistats de targetes i directoris obtinguts de les crides a servidor. Cada tipologia d'objecte és guardada en un arxiu diferent.

Realm permet l'execució de consultes per obtenir els documents guardats anteriorment. Per a això només fa falta especificar la base de dades, les condicions a complir i la classe destí que farà de recipient de les dades.

```
Realm realm = Realm.getInstance(this, type.name() + ".realm");

String parentSearch = (currentFolder == null) ? "" : currentFolder;
RealmResults<? extends RealmObject> result = realm.where(type.getRealmClass())
    .equalTo("parent", parentSearch)
    .findAll();

List<AbstractItem> itemList = Lists.newArrayList();

for(RealmObject card: result) {
    if(Type.CARD.equals(type)) {
        itemList.add(new Card().copy((CardRealm)card));
    }else {
        itemList.add(new Folder().copy((FolderRealm)card));
    }
}
```



### 6.2.11 Motor d'OCR

La digitalització de targetes en l'aplicació és realitzada mitjançant un motor òptic de reconeixement de caràcters, OCR.

Donades les característiques del projecte s'ha optat per utilitzar la llibreria **Tesseract-ocr**. Llibreria open-source desenvolupada originalment per HP i actualment mantinguda per Google.

Tesseract permet aplicar el motor OCR utilitzant un conjunt de fitxers d'entrenament i patrons. Donat el tamany d'aquests fitxers és necessari copiar-los a la memòria externa del dispositiu.

```
private void loadAssets() {
    tesseractPath = FileManager.mkdir(FileManager.FOLDER_TESSER,
        getApplicationContext());
    Boolean syncRequired = !FileManager.checkIfExists(tesseractPath,
        "tessdata/eng.traineddata");

    if(!assetsLoaded && syncRequired) {
        showProgressDialog("Loading", "Wait while checking OCR files...");
        /* Load OCR assets */
        String baseFolder = FileManager.mkdir(FileManager.FOLDER_BASE,
            getApplicationContext());

        TesseractDataTask tesserTask = new TesseractDataTask(getAssets(),
            baseFolder, new AsyncHandler() {

                @Override
                public void postExecute(Boolean isSuccess, String result) {
                    Log.i(TAG, "LOADING SUCCES");
                    dialog.dismiss();
                    unlockScreenOrientation();
                }

            });
        tesserTask.execute();
        assetsLoaded = true;
    }
}
```

La copia dels fitxers d'entrenament i patrons només es realitza en cas de no existir els fitxers en la memòria externa.

Alhora de digitalitzar la targeta, l'aplicació captarà la imatge i la processarà en background, donat que com s'ha esmentat anteriorment en cap moment podem bloquejar el fil principal de l'aplicació.

El primer pas a realitzar és carregar els fitxers del llenguatge a extreure i la imatge obtinguda. *options.inSampleSize*, permet reduir les mides de la imatge original i així disminuir l'ús de memòria i la probabilitat de causar un java heap space en l'aplicació.

```
final TessBaseAPI baseApi = new TessBaseAPI();
baseApi.init(tesseractPath, "eng");

BitmapFactory.Options options = new BitmapFactory.Options();
options.inSampleSize = 2;

Bitmap bitmap = BitmapFactory.decodeFile(imagePath, options );
```

Un cop carregada la llibreria i la imatge s'ha d'orientar correctament i convertir a escala de grisos.

```
ExifInterface exif;
try {
    exif = new ExifInterface(imagePath);
    int exifOrientation = exif.getAttributeInt(
        ExifInterface.TAG_ORIENTATION,
        ExifInterface.ORIENTATION_NORMAL);
    int rotate = 0;
    switch (exifOrientation) {
        case ExifInterface.ORIENTATION_ROTATE_90:
            rotate = 90;
            break;
        case ExifInterface.ORIENTATION_ROTATE_180:
            rotate = 180;
            break;
        case ExifInterface.ORIENTATION_ROTATE_270:
            rotate = 270;
            break;
    }

    if (rotate != 0) {
        int w = bitmap.getWidth();
        int h = bitmap.getHeight();
        // Setting pre rotate
        Matrix mtx = new Matrix();
        mtx.preRotate(rotate);
        // Rotating Bitmap & convert to ARGB_8888, required by tess
        bitmap = Bitmap.createBitmap(bitmap, 0, 0, w, h, mtx, false);
    }
} catch (Exception e) {
    throw Throwables.propagate(e);
}
bitmap = bitmap.copy(Bitmap.Config.ARGB_8888, true);
```

Finalment, executem el motor de reconeixement i obtenim el text contingut en la targeta.

```
String textResult;  
  
baseApi.setPageSegMode(TessBaseAPI.OEM_TESSERACT_CUBE_COMBINED);  
baseApi.setImage(bitmap);  
textResult = baseApi.getUTF8Text();
```

## 7 Proves de l'aplicació

Els jocs de proves d'una aplicació són claus alhora de definir el comportament esperat d'un sistema o de realitzar canvis, assegurant-se del manteniment i integritat de les funcionalitats ja registrades.

### 7.1 Servidor

En el desenvolupament de la part de servidor s'han implementat un conjunt de tests unitaris mitjançant Mocha, funcionament i implementació detallada en la secció 7.2.5.

S'ha decidit testejar la identificació de l'usuari i la validació dels models de dades alhora que la restApi.

### 7.2 Joc de proves

Pel que fa al desenvolupament de l'aplicació Android, s'ha escrit un conjunt de joc de proves a passar en cada iteració de modificació. El joc de proves es va incrementant a mida que s'afegeixen noves funcionalitats. L'addició d'una nova funcionalitat ha de respectar el funcionament de les anteriors.

Nota: Proves realitzades amb el dispositiu Galaxy S3 GT-I9300.

Prova	Passos	Resultat
Identificació en l'aplicació mitjançant Facebook	1. Obrir aplicació 2. Clickar "Log in with Facebook"	Es mostra la pantalla de Facebook i s'identifica l'usuari. L'usuari és dirigit a la pàgina de contingut.
Accés a pantalla de contingut	1. Identificació en l'aplicació o retorn d'una altre pantalla.	Es mostra el llistat de directoris i targetes pertanyents a l'usuari i que es troben en el directori pare.
Accés targeta	1. Seleccionar targeta del llistat	Es mostra la pantalla d'informació de la targeta seleccionada
Accés directori	1. Seleccionar directori	S'accedeix al directori i es mostra el llistat de directoris i targetes contingudes en el directori seleccionat.
Cerca	1. Seleccionar cercar	S'accedeix a la pantalla de cerca.
Introducció de cerca	1. Introducció de contingut a cercar	Es mostren els usuaris i targetes relacionats amb la cerca.
Navegació entre usuaris	1. Es cerca un usuari 2. Es selecciona un usuari del resultat de la cerca	Es mostra les targetes i directoris situats al directori pare de l'usuari.

Importar targeta	<ol style="list-style-type: none"> <li>1. Es navega dins d'un usuari</li> <li>2. Es selecciona una targeta de l'usuari</li> <li>3. Es pitja el botó d'importació</li> </ol>	La targeta seleccionada és copiada a la carpeta <i>Library</i> que es troba al directori pare de l'usuari.
Creació de targeta	<ol style="list-style-type: none"> <li>1. Selecciona el botó de crear targeta</li> <li>2. Introducció de les dades</li> </ol>	El sistema registra la creació de la targeta dins del directori actual.
Creació de directori	<ol style="list-style-type: none"> <li>1. Selecció del botó de crear directori.</li> <li>2. Introducció del nom del directori</li> </ol>	El sistema crea el directori dins del directori actual.
Edició de targeta	<ol style="list-style-type: none"> <li>1. Selecció de targeta</li> <li>2. Es pitja el botó d'edició</li> <li>3. Es modifica informació de la targeta</li> </ol>	El sistema valida els camps de la targeta i en cas afirmatiu registra l'operació. En cas negatiu marca els camps erronis.
Eliminació de la targeta	<ol style="list-style-type: none"> <li>1. Selecció de targeta</li> <li>2. Selecció d'esborrar targeta</li> <li>3. Confirmació d'eliminació</li> </ol>	El sistema registra l'acció i eliminar la targeta seleccionada.
Escanejar targeta	<ol style="list-style-type: none"> <li>1. Selecció opció d'escanejar</li> <li>2. Capura d'imatge</li> <li>3. Selecció del text que conforma cada camp de la imatge</li> <li>4. Modificació dels camps dessitjats</li> </ol>	El sistema valida els camps de la targeta i en cas afirmatiu registra l'operació. En cas negatiu marca els camps erronis.
Eliminació de directori	<ol style="list-style-type: none"> <li>1. Selecciona eliminar directori</li> <li>2. Confirma l'acció d'eliminació</li> </ol>	El sistema elimina el directori i les targetes incloses. El sistema no ha de permetre l'eliminació de directoris amb subdirectoris existents.

El conjunt proves ha de ser suficient per explicar el funcionament bàsic de l'aplicació donat que ha d'assegurar-se la satisfacció de tots els casos detallats en la *secció Casos d'ús 5.8*.

## 8 Planificació

### 8.1 Fases

La gestió del projecte s'ha dividit en quatre grups de processos: inici, planificació, execució i tancament. En el transcurs s'anirà completant la memòria del projecte.

#### 8.1.1 Fase inicial

Definició del projecte. Estudi de solucions existents en el mercat i avaluació de necessitats.

#### 8.1.2 Planificació

Acotació dels objectius a assolir. Definició de les tasques a executar, estimació d'hores i avaluació de possibles riscos existents en l'execució del projecte.

#### 8.1.3 Execució

La fase d'execució es centra en el disseny i implementació del producte. Consta de les següents iteracions:

Funcionalitats	Iteració
Desenvolupament de backend i model de dades. Validació d'informació a guardar i API per una correcta interacció.	1
Pàgina web. Validació de la API mitjançant la interacció de l'usuari amb la web. Identificació utilitzant sdk de Facebook. Introducció i cerca de targetes.	2
Posada en funcionament aplicació mòbil. Identificació de l'usuari al sistema utilitzant token de facebook i credencials del backend.	3
Incorporació de motor OCR per a la digitalització de les targetes en l'aplicació mòbil.	4

La finalització de cada iteració comporta la obtenció d'un producte sòlid que compleix amb les necessitats del projecte.

#### 8.1.4 Tancament

Fase consistent en validar el producte final obtingut i assegurar un mínim de qualitat alhora que s'assoleixen els objectius marcats en el projecte.

Finalitza amb la revisió de la memòria i documentació del projecte.

## 8.2 Anàlisi de riscos

En aquest apartat s'analitzen els principals riscos que poden afectar el projecte amb la finalitat de raonar sobre l'impacte que aquests poden tenir i elaborar un pla de contingència.

El conjunt de riscos analitzats són agrupats en tres blocs diferents en funció del seu origen: riscos dependents del desenvolupament del projecte (**PRO**); riscos dependents de les persones físiques (**PF**); i finalment, riscos provinents del propi sistema (**S**).

Codi	Probabilitat	Impacte	Descripció
<b>R1-PRO</b>	2/10	7/10	Previsió temporal inadequada, incompliment de terminis d'entrega.
<b>R2-PRO</b>	4/10	5/10	Canvis inesperats en la definició del producte. Canvis d'implementacions o funcionalitats donada la dificultat o impossibilitat d'execució dels definits en primera instància.
<b>R3-PRO</b>	1/10	9/10	Qualitat insuficient, el producte no compleix el resultat esperat i no cobreix les funcionalitats correctament.
<b>R4-PF</b>	8/10	2/10	Mala assignació dels camps extrets de les targetes mitjançant OCR amb els corresponents camps d'informació.
<b>R5-PF</b>	2/10	3/10	Usuari perd les credencials d'accés al sistema
<b>R6-PF</b>	2/10	3/10	Pèrdua del mòbil, o esborrat complet del sistema
<b>R7-S</b>	1/10	9/10	Caiguda del servidor

A continuació, es proposen el conjunt d'accions a realitzar en funció dels casos esmentats a l'anterior taula. Donat que no és possible determinar a priori quan succeiran els fets i la probabilitat és força subjectiva, el conjunt d'accions proposades té com a finalitat la reducció de l'impacte dels riscos en el correcte funcionament del sistema.

Codi	Contingència	Acció
<b>R1-PRO</b>	-	Assignació de més recursos, limitat abast del projecte i replanificació del temps estimat.
<b>R2-PRO</b>	Acotament correcte i detallat de les funcionalitats a implementar en el projecte.	Avaluar l'impacte del canvi i la viabilitat d'execució dins del pla establert. En cas positiu, dur a terme el canvi, altrament desestimar.
<b>R3-PRO</b>	Cerca d'informació, projectes semblants, tutorials i altres fonts relacionades.	Tornar a l'especificació de la funcionalitat afectada. Avaluació dels errors i proposta de solucions viables. Replanificació del temps estimat i tasques implicades.

<b>R4-PF</b>	Aplicació de Regex per processar la informació.	Oferir la possibilitat de reassignar els camps.
<b>R5-PF</b>	Utilitzar validació de credencials externs com és Facebook o Google+.	-
<b>R6-PF</b>	Validació i registre de totes les accions al servidor.	-
<b>R7-S</b>	Descentralitzar l'aplicació.	Utilitzar més d'un servidor per oferir el servei

### 8.3 Estimació d'hores

A continuació es detalla l'estimació d'hores que ha comportat el projecte. Remarcar que és una *estimació* donat que l'execució del projecte s'ha compaginat amb el món laboral i conseqüentment és difícil acotar de manera exacte les hores destinades a cada fase.

<b>Fase inicial (60h)</b>	
Definició del projecte	20
Estudi de solucions existents en el mercat	20
Avaluació necessitats	10
Redacció memòria	10
<b>Planificació (40h)</b>	
Planificació i anàlisi de requisits	30
Redacció memòria	10
<b>Execució (390h)</b>	
Disseny de l'aplicació	50
Implementació	280
Proves	40
Redacció memòria	20
<b>Tancament (45h)</b>	
Retocs	30
Manual d'usuari	5
Finalitzar memòria	10
<b>Total: 535h</b>	



## 8.4 Cost econòmic

En aquesta secció s'exposa una simulació del cost econòmic que tindria el projecte en àmbit laboral portat per un sol programador.

- **Domini:** 12€ (*any*)
- **Servidor OVH,** 3 cores, 4GB de Ram i 50GB de disc en Raid 10: 116 € (*anual*).
- **Llicència Desenvolupador Android:** 25€

**Cost recursos humans:** Programador (10€/h) – Analista (20€/h)

<b>Programador (375h)</b>		
Redacció documents	20h	200€
Manual d'usuari	5h	50€
Implementació	220h	2.200€
Proves	40h	400€
Retocs	30h	300€
Formació	60h	60€
<b>Analista (160h)</b>		
Especificació	30h	600€
Redacció documents	20h	400€
Definició del producte	20h	400€
Disseny de l'aplicació	50h	1.000€
Finalitzar documents	10h	200€
Redacció documents	10h	200€
Planificació i anàlisi	20h	400€
	<b>535h</b>	<b>6.410€</b>

Cost total del projecte: **6.563€**

## 9 Conclusions

L'estudi i desenvolupament d'un producte des de zero no acostuma ser una tasca fàcil. S'han d'estudiar els motius i/o necessitats que porten a voler realitzar-lo, així com els detonants socials causants d'aquests requisits.

Acostuma a ser freqüent que pensem idees o solucions als nostres problemes i volem aplicar la primera solució que se'ns passa pel cap i no som conscients que potser altra gent s'ha trobat prèviament amb el problema i ja han realitzat una proposta prou interessant o han arribat a unes conclusions que podrien ésser de profit com a punt de partida pel nostre producte.

La idea de fer una aplicació per gestionar les targetes de contacte va sorgir d'anar a diverses xerrades d'emprenedoria on tothom frisava per repartir les targetes o per col·leccionar-les. Masses targetes on no poder apuntar notes o no poder classificar-les com era adient.

La realització d'aquest projecte ha estat satisfactòria donat que ha permès la posada en pràctica de coneixements adquirits al llarg de la carrera, així com l'obtenció d'un producte arran de les meves idees. També ha permès profunditzar en noves tecnologies com NodeJs i AngularJs.

Fer un producte que ha d'interactuar amb un usuari final fa veure la importància de tenir un bon dissenyador al darrere que especifiqui la paleta de colors a utilitzar o que ens proveeixi de les icones i imatges correctes. La posada en marxa d'un producte al món laboral no només és part tecnològica sinó que interactuen diverses disciplines i això fa que durant la realització del projecte s'hagin anat plantejant dubtes de com gent externa les duria a terme (comunicació, imatge de marca, interacció, llenguatge, model de negoci).

És cert, que durant el procés van sorgint noves idees i noves funcionalitats que m'hauria agradat poder incloure en el producte final, com és l'intercanvi de targetes personals per NFC o un sistema de recomanacions en funció dels esdeveniments guardats d'un usuari a causa de l'interacció d'aquest amb el sistema, però per realitzar el projecte final de carrera era necessari acotar les tasques i deixar funcionalitats per en un futur amb més tranquil·litat provar de posar-les en pràctica.

## 10 Treball futur

En aquesta secció s'esmenten possibles millores o tasques a dur a terme un cop finalitzat el projecte.

### 10.1 Monetització

Una tasca important a dur a terme seria buscar un model de negoci apte per poder tirar endavant econòmicament la idea. Un exemple seria monetitzar la digitalització de targetes mitjançant un serveix extern d'OCR, donat que la llibreria utilitzada tenia forces mancances, però era la única mínimament decent en el camp d'open source.

S'hauria d'estudiar l'ús que en fan els usuaris extraient mètriques i després veure com aplicar monetització a l'aplicació.

Un pla de negoci correcte necessita d'un estudi detallat i de varies propostes.

### 10.2 Comunicació de camp proper

S'ha estudiat la idea d'incloure NFC tecnologia per transmetre imatges entre dispositius.

#### 10.2.1 Què és la tecnologia de Comunicació de camp proper - NFC?

La Comunicació de camp proper o NFC (*Near Field Communication*) és una tecnologia **estandaritzada** que té com a propòsit facilitar la **interconnexió** de dispositius i **l'intercanvi** de dades en un entorn acotat. Aquesta tecnologia es basa en la identificació per radiofreqüència (**RFID**), una tecnologia **sense fils** que permet l'ús d'un objecte que s'afegeix a un segon objecte amb el propòsit d'identificació i seguiment utilitzant ones de ràdio.

El RFID consta de dues parts principals: un circuit integrat que emmagatzema i processa la informació, modula i desmodula el senyal de radiofreqüència; i d'una antena per rebre i transmetre el senyal. En canvi, la tecnologia NFC combina ambdues funcions RFID en un únic dispositiu, conservant la **compatibilitat** amb tota la infraestructura existent de RFID.

NFC es basa en el principi d'inducció electromagnètica pel qual dos circuits inductius propers comparteixen energia al ser col·locats dins els seus respectius camps propers podent així transmetre dades a distàncies de pocs centímetres, distància que oscil·la dels **20cm** als **4cm** amb les antenes estàndard.

A més, els dispositius NFC són capaços de **rebre** i **transmetre** dades al mateix temps i, com a conseqüència, poder **detectar possibles errors** si la freqüència del senyal rebut no coincideix amb la freqüència de senyal transmès.

### 10.2.2 Estudi de sistemes NFC

A continuació es detalla el conjunt de sistemes informàtics existents que utilitzen la tecnologia NFC per transmetre o connectar-se entre diversos dispositius.

#### Blue NFC

-	Empresa gestora
Internacional	Àmbit polític
Quotidià	Sectors
Android: 10.000 a 50.000 d'usuaris.	Quota de mercat

#### Serveis

L'aplicació ofereix la possibilitat de connectar-se amb altres dispositius mitjançant Bluetooth utilitzant com a medi d'establiment de credencials la tecnologia NFC.

#### Proxama - NFC Mobile Wallet & Marketing

<a href="http://www.proxama.com">www.proxama.com</a>	Adreça web
Proxama	Empresa gestora
Internacional	Àmbit polític
Negocis, Empresarial	Sectors

#### Serveis

La companyia ofereix diversos serveis referents a la tecnologia NFC. Entre ells es troben la possibilitat de fer transaccions mitjançant el mòbil amb les quals fer pagaments, a més, inclou PIN de seguretat per evitar i/o disminuir possibles riscos de robatori d'informació.

Els serveis pertanyents a la vessant de màrqueting consisteixen en facilitar diversos punts estratègics mitjançant els quals obtenir informació o realitzar accions amb el simple ús del mòbil i de tags NFC.

#### Objectius

L'objectiu principal de la companyia és realitzar transferències segures mitjançant contactless i oferir a les marques, agències i companyies de comunicació una manera eficient i ràpida d'obtenir i premiar als consumidors.

#### Forma de negoci

Proxama és Partnership de Mastercard amb la qual va llançar la primera aplicació mòbil per donar servei al pagament contactless. A més, també és Partnership de Trustonic amb la qual treballa per oferir la següent generació de seguretat m-commerce.

La principal fons d'ingressos és la realització de campanyes de màrqueting d'aproximació, fins al moment més de 600.

### GoToTags

<a href="http://www.gototags.com">www.gototags.com</a>	Adreça web
Wireless Sensor Technologies, Inc.	Empresa gestora
Internacional	Àmbit polític
Negocis, Empresarial	Sectors

### Serveis

Les funcionalitats ofertes per GoToTags són la possibilitat de configurar tags NFC per a ser utilitzats en altres aplicacions de propòsits personals o amb finalitats comercials i de màrqueting. La configuració dels tags es realitza mitjançant una aplicació d'escriptori desenvolupada per ser executada en Windows.

### Forma de negoci

Wireless Sensor Technologies, Inc. basa el model de negoci en l'ofertament gratuït de l'aplicació per configurar els tags. El maquinari i els diferents objectes on incorporar la tecnologia NFC són de pagament.

### NFC Launcher

<a href="http://www.nfclauncher.com">www.nfclauncher.com</a>	Adreça web
TIMWE	Empresa gestora
Internacional	Àmbit polític
Negocis	Sectors

### Serveis

Les funcionalitats ofertes per NFC launcher consisteixen en: configuració de tags per a mostrar informació o realitzar accions específiques; i, aplicació Android encarregada de rebre la informació o ordre dels tags i executar-la correctament. A més, també ofereixen el servei de customització i suport en l'adaptació de campanyes de màrqueting creant un canal únic i personalitzat.

### Forma de negoci

El model de negoci de TIMWE consisteix en oferir l'aplicació de manera gratuïta i oferir el serveis de campanyes de màrqueting de pagament.

### 10.2.3 Comparativa - Tecnologia NFC

A continuació es prossegueix a presentar la comparativa de les diverses funcionalitats i aspectes d'interès que s'han observat en els anàlisis de les empreses i projectes anteriors referent a la utilització de la tecnologia NFC en sistemes d'informació.

S'ha observat com les diverses solucions existents estan orientades a realitzar accions determinades en els mòbils receptors o oferir informació específica. A més, algunes d'aquestes ofereixen la tecnologia NFC com una eina per realitzar màrqueting de manera personalitzada podent efectuar un millor seguiment en dit canal.

Un punt a tenir en compte és la utilització del NFC com a mètode segur per establir una connexió posterior, en els cassos anteriors, per iniciar una connexió Bluetooth entre dos dispositius o per efectuar de manera segura el pagament.

### 10.3 Recomanacions

És usual que els usuaris es focalitzin en cert grup d'interessos o en un conjunt de temàtiques concretes. Així doncs, seria interessant tant a nivell pràctic com a nivell personal d'aprenentatge, mirar d'incloure un sistema de remanacions en funció de l'interacció de l'usuari amb el sistema.

Accions com etiquetar una targeta, escriure una anotació o realitzar una cerca diuen molt de les preferències d'un usuari.

La idea seria realitzar un procés batch on anar calculant aquest conjunt de recomanats per cada usuari. Això seria possible i tindria el seu valor afegit si l'aplicació tingués un gran nombre d'usuaris actius.

# 11 Bibliografia

## Articles

- Estudi ús d'aplicacions: <http://www.itespresso.es/el-camino-de-una-app-movil-49628.html>
- Cuota de mercat Android (2/10/2013) <http://www.xatakandroid.com/mercado/mas-de-la-mitad-de-los-dispositivos-android-llevan-jelly-bean>

## Gestors de contactes:

- CardMunch de LinkedIn: <http://www.cardmunch.com/>
  - Info: <http://www.crunchbase.com/company/cardmunch>
  - <http://www.how-to-really-use-linkedin.com/uncategorized/linkedin-cardmunch-eats-your-business-cards/>

NFC: [http://en.wikipedia.org/wiki/Near\\_field\\_communication](http://en.wikipedia.org/wiki/Near_field_communication)

RFID: [http://en.wikipedia.org/wiki/Radio-frequency\\_identification](http://en.wikipedia.org/wiki/Radio-frequency_identification)

OCR: [http://en.wikipedia.org/wiki/Optical\\_character\\_recognition](http://en.wikipedia.org/wiki/Optical_character_recognition)

- Tutorials: <http://gaut.am/making-an-ocr-android-app-using-tesseract/>
  - <http://rmtheis.wordpress.com/2011/08/06/using-tesseract-tools-for-android-to-create-a-basic-ocr-app/>
- Llibreria: <https://code.google.com/p/tesseract-ocr/>
  - <https://code.google.com/p/tesseract-android-tools/>
- Servei on cloud: <http://ocrapiservice.com/rates/>
  - <https://github.com/smart-mobile-software/ocrapiservice>

## Servidor

- Iniciar servidor [www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-14-04](http://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-14-04)
- Nginx: [www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-14-04-lts](http://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-14-04-lts)
- Host Records: [www.namecheap.com/support/knowledgebase/article.aspx/716/62/all-host-records](http://www.namecheap.com/support/knowledgebase/article.aspx/716/62/all-host-records)

## Facebook

- OAuth: <http://passportjs.org/guide/oauth-api/>
- Graph-api: <https://developers.facebook.com/docs/graph-api/reference/v2.2>
- facebook-node: <https://github.com/Thuzy/facebook-node-sdk/>

## Android

- Realm: <http://realm.io/>
- AndroidListViews: <http://www.vogella.com/tutorials/AndroidListView/article.html>

- Copiar assets: <http://www.twodee.org/blog/?p=4518>
- Async Http Client: <http://loopj.com/android-async-http/>

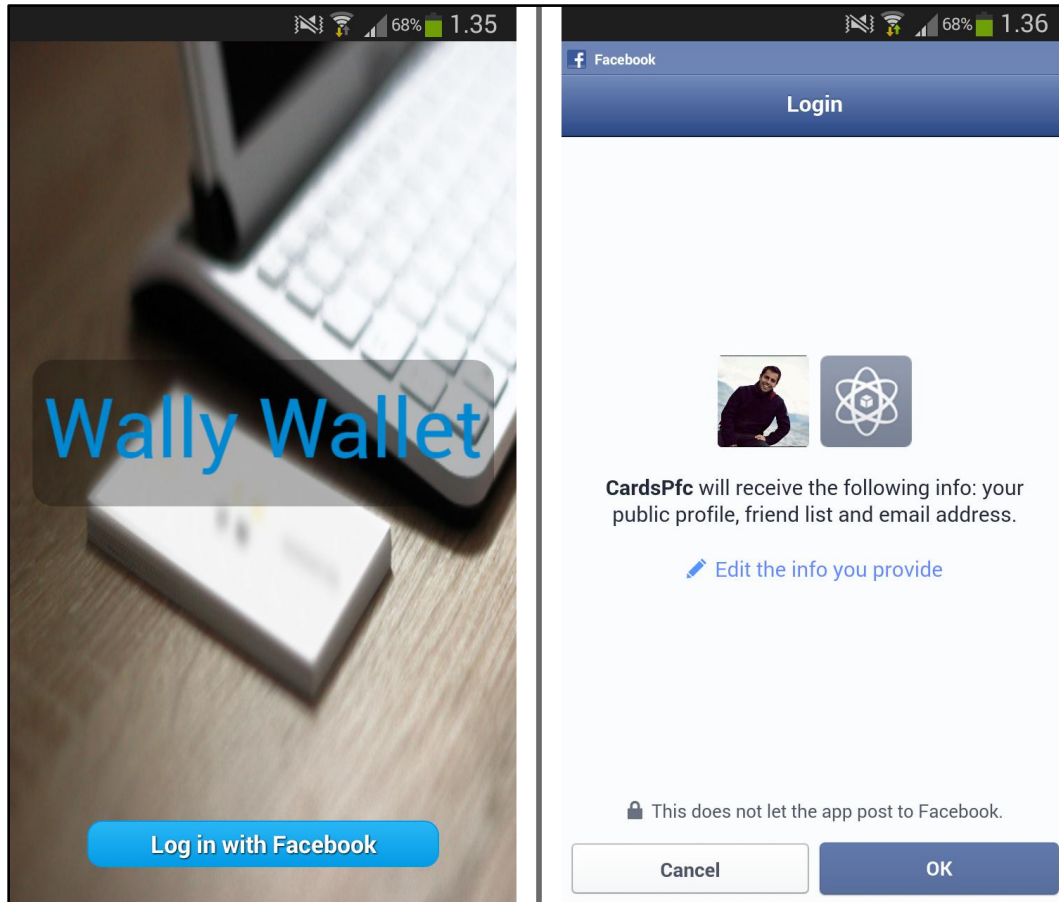
## Recursos Web

- AngularJs: <https://docs.angularjs.org/api/>
- Bootstrap: <http://getbootstrap.com/>
- Angular + Bootstrap: <http://angular-ui.github.io/bootstrap/>
- MoongODB: <http://www.mongodb.org/>
- Mongoose: <http://mongoosejs.com/docs/guide.html>
- Mongoose-schema-extend: <https://github.com/briankircho/mongoose-schema-extend>
- Mocha: <http://mochajs.org/>
- NodeJs authentication: <http://scotch.io/tutorials/javascript/easy-node-authentication-setup-and-local>



## 12 Manual d'usuari

En aquesta secció es detalla el manual d'usuari per a l'aplicació Android. S'anirà exposant les diverses pantalles que podem trobar i quines accions hi ha disponibles en cadascuna.



Imatge 7: Inici i credencials de Facebook

### Pantalla d'inici

Pantalla inicial que ens trobem a l'iniciar l'aplicació. Ens demanarà d'iniciar la sessió amb l'usuari de facebook.

### Credencials Facebook

En cas de ser el primer cop que entrem a l'aplicació, facebook ens demanarà acceptar les condicions d'ús del sistema.

Les credencials sol·licitades són accés de lectura al perfil, llista d'amistats i adreça electrònica.



**Imatge 8: Pantalla inicial i de cerca**

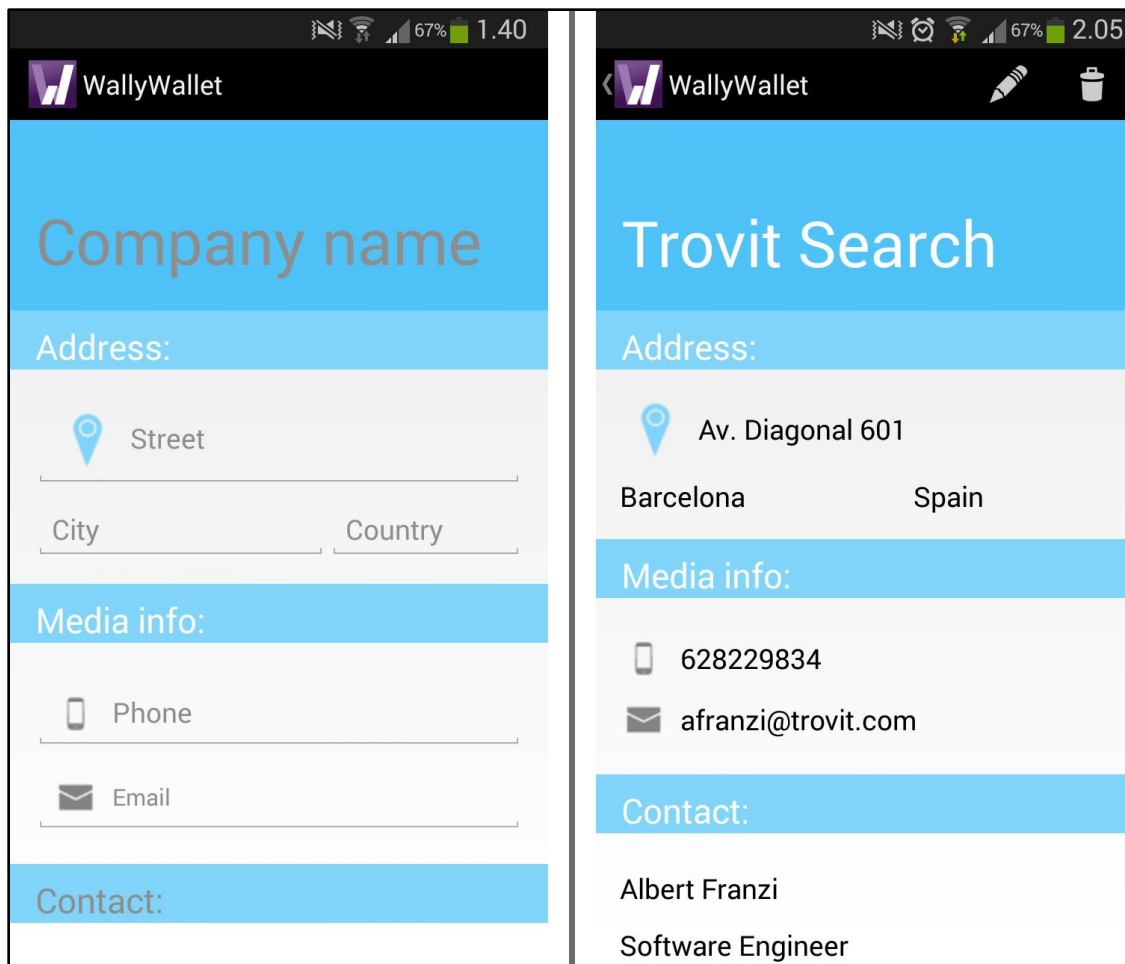
Un cop identificats, l'aplicació ens portarà a la pantalla de contingut on se'ns mostrarà el directori principal amb les seves targetes i subdirectoris.

Les opcions disponibles en la pantalla inicial són cercar usuaris i targetes i escanejar targeta física (*imatge 12*); crear targeta nova i seleccionar targeta (*imatge 11*); crear nou directori; seleccionar directori.

### **Cercar**

En la pantalla de cerca, tenim un camp per introduir el text a cercar i tres tipus de cerca: usuaris, targetes o etiquetes de targetes. Per cada usuari se'ns informarà del nombre de targetes i directoris que té registrats.

Al seleccionar un usuari s'obrirà la pantalla inicial de contingut de l'usuari. En cas de seleccionar una targeta serem redirigits a la pantalla d'informació de la targeta seleccionada.



Imatge 9: Edició i consulta de targeta

### Crear de targeta

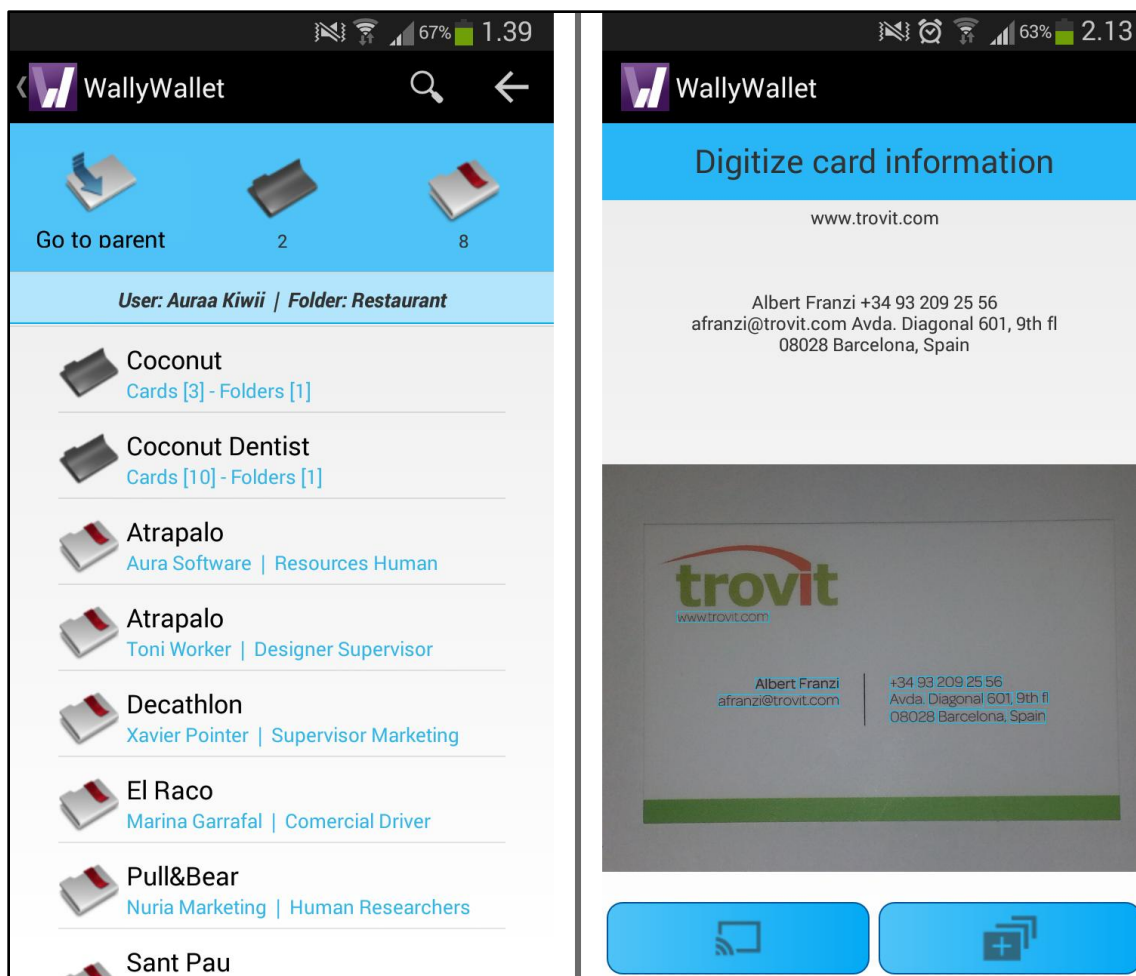
La pantalla de creació ens permet introduir els camps necessaris d'una nova targeta de contacte. Tenim com a camps disponibles: companyia, adreça, telèfon i correu electrònic, nom de contacte i rol dins l'empresa. Un cop introduïts els camps pitgem el botó de guardar i el sistema registrarà la nova targeta al nostre compte.

En cas de tenir algun camp erroni ens informará amb un missatge, indicant quin camp està malament.

### Consultar targeta

Referent a la pantalla de consulta, se'ns mostra la informació de la targeta seleccionada prèviament. Podem editar el contingut o esborrar la targeta. El sistema demanarà confirmació en aquesta última acció.

En cas d'estar consultant la targeta d'un altre usuari, podrem importar-la al nostre directori de targetes.



Imatge 10: Navegació en directoris i escanejat de targetes

## Navegació

Quan seleccionem un directori, se'ns mostrarà el conjunt de targetes i directoris continguts. En la pantalla podem veure el nombre de directoris (2) i targetes (8) que hi ha en el directori Restaurant. En cas d'estar consultant el directori d'un altre usuari, també se'ns informarà de quin usuari és.

Per tornar enrere, podem pitjar el botó de *Go to parent* o simplement el botó *back* del dispositiu mòbil.

## Escanejat

En aquesta part l'aplicació activarà la càmera del dispositiu per fer una fotografia a la targeta. Un cop confirmada que la targeta és vàlida, extraurà el text contingut.

## 13 Glossari

**Smartphone:** Es tracta d'un telèfon mòbil construït sobre un sistema operatiu mòbil, amb capacitat de computació avançada i connectivitat que permet a l'usuari instal·lar aplicacions per a qualsevol utilitat.

**Cookie:** Fragment d'informació enviat des d'un servidor de pàgines web a un navegador. El navegador guarda aquesta informació en forma d'arxiu de text al disc dur del visitant de la pàgina web per tal que certes informacions puguin ser recuperades en posteriors visites.

**IDE:** Un entorn integrat de desenvolupament (en anglès, *Integrated Development Environment*) és una eina informàtica per al desenvolupament de programari de manera còmoda i ràpida, facilitant la tasca del programador.

**Proxy:** Serveix per a permetre l'accés a Internet a tots els equips d'una organització quan només es pot disposar d'un únic equip connectat, això és, una única adreça IP.

**CRM:** Customer relationship management. És un software destinat a gestionar la informació dels clients d'empreses o altres tipus d'organitzacions. Registra la informació dels clients/contactes, a nivell individual i d'organització, i les seves interrelacions.

**Business Use Case (BUC):** Manera en la qual un client o alguna altra part interessada pot fer ús del producte per aconseguir el resultat que volen.

**Object-relational Mapping (ORM):** és una tècnica de programació per convertir dades de llenguatges de programació orientats a objectes en la seva representació en bases de dades relacionals, a través de la definició de les correspondències entre els diferents sistemes.

